Exhibit A
to
Stipulation Re: Visto's Filing of First Amended Answer and Counterclaims

MANATT, PHELPS & PHILLIPS, LLP
ROBERT D. BECKER (Bar No. CA 160648)
E-mail: rbecker@manatt.com
EUGENE L. HAHM (Bar No. CA 167596)
E-mail : ehahm@manatt.com
SHAWN G. HANSEN (Bar No. CA 197033)
E-mail: shansen@manatt.com
1001 Page Mill Road, Building 2
Palo Alto, CA 94304-1006
Telephone: (650) 812-1300
Facsimile: (650) 213-0260

Attorneys for Defendant
and Counterclaimant
VISTO CORPORATION

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

RESEARCH IN MOTION LIMITED, Plaintiff, vs. VISTO CORPORATION, Defendant.	Case No. C-07- 3177 (MMC) **FIRST AMENDED ANSWER TO RESEARCH IN MOTION LIMITED'S COMPLAINT FOR PATENT INFRINGEMENT; COUNTERCLAIMS AGAINST RESEARCH IN MOTION LIMITED AND RESEARCH IN MOTION CORPORATION** **JURY TRIAL DEMANDED**
VISTO CORPORATION, Counterclaimant, vs. RESEARCH IN MOTION LIMITED, and RESEARCH IN MOTION CORPORATION, Counterdefendants.	

Defendant Visto Corporation ("Visto") hereby submits this First Amended Answer and Counterclaims in response to the Complaint for Patent Infringement of U.S. Patent No. 5,889,839 ("the '839 patent") brought by Plaintiff Research In Motion Limited ("RIM Ltd.") as follows:

NATURE OF THE ACTION

1. Answering paragraph 1 of the Complaint, Visto admits that RIM Ltd.'s Complaint purports to be a civil action for patent infringement arising under 35 U.S.C. §1, *et seq*. Except as expressly admitted, Visto denies the remaining allegations of paragraph 1.

PARTIES

2. Answering paragraph 2 of the Complaint, Visto admits upon information and belief that Plaintiff RIM Ltd. is a Canadian corporation with its principal place of business at 295 Phillip Street, Waterloo, Ontario, Canada N2L 3W8.

3. Answering paragraph 3 of the Complaint, Visto admits that it is a corporation organized and existing under the laws of the State of Delaware and that its principal place of business is at 101 Redwood Shores Parkway, Redwood City, California 94065.

JURISDICTION

4. Answering paragraph 4 of the Complaint, Visto admits that this Court has subject matter jurisdiction for claims under the Patent Laws of the United States, Title 35, United States Code pursuant to 28 U.S.C. §§ 1331 and 1338(a).

5. Answering paragraph 5 of the Complaint, Visto admits that its contacts are sufficiently continuous and systematic to constitute doing business within the State of California and within this District. Visto has established its principal place of business within the State of California and within this District at 101 Redwood Shores Parkway in Redwood City. This Court has personal jurisdiction over Visto. Visto admits that it offers services and/or products under the name "Visto Mobile." Except as expressly admitted, Visto denies the remaining allegations of paragraph 5.

VENUE

6. Answering paragraph 6 of the Complaint, Visto admits the allegations of paragraph 6.

INTRADISTRICT ASSIGNMENT

7. Answering paragraph 7 of the Complaint, Visto admits that this action arises under the patent statutes and therefore, according to Civil Local Rule 3-2(c), it may be assigned on a district-wide basis.

FACTUAL BACKGROUND

8. Answering paragraph 8 of the Complaint, Visto is without sufficient information to form a belief as to the truth of the allegations contained therein, and therefore denies each and every allegation contained therein.

9. Answering paragraph 9 of the Complaint, Visto is without sufficient information to form a belief as to the truth of the allegations contained therein, and therefore denies each and every allegation contained therein.

10. Answering paragraph 10 of the Complaint, Visto admits that a copy of what is purportedly United States Patent No. 5,889,839 is attached as Exhibit A to the Complaint. Visto is without knowledge or information sufficient to form a belief as to the truth of the remaining allegations of paragraph 10, and on that basis denies the remaining allegations of paragraph 10.

11. Answering paragraph 11 of the Complaint, Visto admits the allegations of paragraph 11.

12. Answering paragraph 12 of the Complaint, Visto admits that Visto Mobile™ is the brand name for Visto's patented comprehensive architecture and product suite that provides continuously updated, secure and reliable mobile access to corporate and personal email, calendar and address book data on all popular mobile devices and networks. Visto Mobile™ comprises server, desktop, and device software components that are integrated with the Visto Mobile Access Platform. Except as expressly admitted, Visto denies the remaining allegations of paragraph 12.

13. Answering paragraph 13 of the Complaint, Visto denies each and every allegation contained therein.

COUNT I

(Infringement of the '839 Patent)

14. Answering paragraph 14 of the Complaint, Visto incorporates by reference each

and every allegation contained in Paragraphs 1-13 of its Answer.

15. Answering paragraph 15 of the Complaint, Visto denies each and every allegation contained therein.

16. Answering paragraph 16 of the Complaint, Visto denies each and every allegation contained therein.

17. Answering paragraph 17 of the Complaint, Visto denies each and every allegation contained therein.

18. Answering paragraph 18 of the Complaint, Visto denies each and every allegation contained therein.

RESPONSE TO RIM'S PRAYER FOR RELIEF

19. Visto denies that RIM is entitled to any relief whatsoever in connection with its Complaint, including, but not limited to, all relief requested in RIM Ltd.'s Prayer for Relief, paragraphs A-G.

20. Except as expressly admitted herein, Visto denies all factual allegations set forth in RIM Ltd.'s Complaint.

VISTO'S AFFIRMATIVE DEFENSES

Visto asserts the following affirmative defenses against RIM Ltd.'s patent infringement claims:

First Affirmative Defense (Failure to State a Claim)

21. The Complaint fails to state a claim upon which relief may be granted.

Second Affirmative Defense (No Infringement)

22. Visto has not and does not willfully or otherwise directly infringe, contribute to the infringement of, or actively induce others to infringe any claim of the '839 patent.

Third Affirmative Defense (Patent Invalidity)

23. The claims of the '839 patent are invalid for failure to comply with the requirements of 35 U.S.C. §§ 101, 102, 103, and/or 112.

Fourth Affirmative Defense (Prosecution History Estoppel)

24. By reason of the proceedings in the United States Patent and Trademark Office

during the prosecution of the '839 patent, RIM Ltd. is estopped from claiming a construction of the patent claims that would cover or include, either literally or by application of the Doctrine of Equivalents, any apparatus, device, product, or method made, used, sold, or offered for sale by Visto.

Fifth Affirmative Defense (Limitation by Prior Art)

25. The prior art known before the alleged inventions so limits and restricts the scope of the claims of the '839 patent that Visto cannot be considered to have infringed the '839 patent.

Sixth Affirmative Defense (Limitation on Damages)

26. RIM Ltd.'s claims for damages and injunction are barred in whole or in part by the provisions of 35 U.S.C. §§ 271, 286, and/or 287.

Seventh Affirmative Defense (Laches)

27. The claims asserted in the Complaint are barred by the doctrine of laches.

Eighth Affirmative Defense (Waiver)

28. . The claims asserted in the Complaint are barred by the doctrine of waiver.

Ninth Affirmative Defense (Estoppel)

29. The claims asserted in the Complaint are barred by the doctrine of estoppel.

30. Visto reserves the rights to assert any other basis for invalidity, unenforceability, or any other defense that discovery may reveal.

PRAYER FOR RELIEF

WHEREFORE, Visto respectfully requests the Court to enter judgment in Visto's favor as to all claims asserted against it in RIM Ltd.'s Complaint and, specifically, to enter judgment:

1. Denying all relief sought in RIM Ltd.'s Complaint;

2. Declaring that Visto has not infringed, contributed to, or induced infringement of the claims of the '839 patent.

3. Dismissing RIM Ltd.'s Complaint and action with prejudice;

4. Declaring that the asserted claims of the '839 patent are invalid;

5. Finding that this case is "exceptional" and that Visto is accordingly entitled to an award against RIM Ltd. of its reasonable attorneys fees pursuant to 35 U.S.C. § 285; and

6. Awarding to Visto reasonable costs and disbursements of this action, attorneys' fees, together with interest, and including prejudgment interest.

7. Granting Visto such other and further relief as this Court may deem just and proper.

VISTO'S FIRST AMENDED COUNTERCLAIMS AGAINST RIM LTD. and RESEARCH IN MOTION CORPORATION

Visto alleges as follows for its Counterclaims against RIM:

PARTIES

1. Visto is a corporation organized and existing under the laws of the State of Delaware with a place of business at 101 Redwood Shores Parkway, 4th Floor, Redwood City, California 94065.

2. Research In Motion Limited ("RIM Ltd.") is a corporation organized and existing under the laws of Ontario, Canada with a principal place of business at 295 Phillip Street, Waterloo, Ontario, Canada N2L 3W8. RIM Ltd. is doing business in the Northern District of California and elsewhere in the United States.

3. Research In Motion Corporation ("RIM Corp.") is a corporation organized and existing under the laws of the State of Delaware and registered to do business in the State of Texas with a place of business at 122 West John Carpenter Parkway, Suite 430, Irving, Texas 75039, which is doing business and infringing Visto's patents in the Eastern District of Texas and elsewhere in the United States. Research In Motion Corporation is the United States distributor of Research In Motion Limited's products and services. RIM Ltd. and RIM Corp. are hereinafter referred hereto jointly as "RIM."

JURISDICTION, VENUE & INTRADISTRICT ASSIGNMENT

4. This is a civil action for patent infringement arising under the patent laws of the United States, 35 U.S.C. § 271 et seq.

5. This Court has subject matter jurisdiction over this action pursuant to 28 U.S.C. §§ 1331 and 1338(a).

6. RIM has sufficient contacts with the Northern District of California to subject it to

the personal jurisdiction of this Court for purposes of this Counterclaim, including, without limitation, acts of infringement of Visto's patents committed by RIM within this District. RIM has a pending civil action against Visto in this district and thus has submitted to the jurisdiction in this district.

7. Venue is proper in this District pursuant to 28 U.S.C. § 1391(b)-(d) and 28 U.S.C. § 1400(b).

8. Pursuant to Civil L.R. 3-2(c), this action for patent infringement shall be assigned on a district-wide basis.

GENERAL ALLEGATIONS

9. Established in 1996, Visto is a leading provider of personal and corporate wireless messaging solutions to mobile operators for personal and corporate use. Visto's inventions enable the deployment of a complete mobility solution, which provides secure access to and synchronization of the most widely used personal information management ("PIM") data over any network and on a broad array of mobile devices, such as personal digital assistants ("PDAs"), smart phones and the like. Visto has expended considerable resources in inventing and developing its inventions and protecting its rights therein.

10. Visto holds all right, title, and interest in and to United States Patent No. 7,225,231, entitled "System And Method For Transmitting Workspace Elements Across A Network" ("'231 patent"), which was duly and properly issued by the United States Patent and Trademark Office ("USPTO") on May 29, 2007 in the name of Daniel J. Mendez et al. A copy of the '231 patent is attached as Exhibit 1 to this First Amended Answer and Counterclaims.

11. Visto holds all right, title, and interest in and to United States Patent No. 7,228,383, entitled "System And Method For Progressive And Hierarchical Caching" ("'383 patent"), which was duly and properly issued by the USPTO on June 5, 2007 in the name of Friedman et. al. A copy of the '383 patent is attached as Exhibit 2 to this First Amended Answer and Counterclaims.

12. RIM provides products and services that cache data to portable devices and that enable access to and/or synchronization of data in secure network environments (collectively, the

"Accused Products"), including without limitation the products and services currently sold by RIM under the name Blackberry.®

13. RIM infringes the '231 and '383 Patents directly, contributorily and/or by active inducement by importing, manufacturing, using, marketing, distributing, selling, and/or supporting the Accused Products.

14. Upon information and belief, RIM's infringement of the '231 and '383 Patents has been and continues to be willful.

COUNT I

(Declaratory Judgment of Non-Infringement)

15. Visto realleges and incorporates by reference the foregoing First Amended Counterclaims paragraphs 1 – 14 as though fully set forth herein.

16. RIM Ltd. alleges in its Complaint that Visto has infringed the '839 patent. Visto denies RIM Ltd.'s allegations with respect to the '839 patent. Therefore, there is a justiciable controversy between Visto and RIM Ltd. concerning Visto's liability for infringement of the '839 patent.

17. Visto has not infringed, either directly, contributorily, or by inducement, any valid and enforceable claim of the '839 patent.

COUNT II

(Declaratory Judgment of Invalidity)

18. Visto realleges and incorporates by reference the foregoing Counterclaim paragraphs 1-17 as though fully set forth herein.

19. RIM Ltd. alleges in its Complaint that Visto has infringed the '839 patent. Visto denies RIM Ltd.'s allegations with respect to the '839 patent. Therefore, there is a justiciable controversy between Visto and RIM Ltd. concerning Visto's liability for infringement of the '839 patent.

20. The '839 patent is invalid for failing to satisfy the conditions of patentability set forth in Title 35 of the United States Code, including, without limitation, the requirements of §§ 101, 102, 103 and 112.

COUNT III

(Infringement of U.S. Patent No. 7,225,231)

21. Visto incorporates paragraphs 1 through 20 as though fully restated herein.

22. RIM has infringed and continues to infringe the '231 patent in this District and elsewhere in the United States by RIM's manufacture, importation, sale, offering for sale, and/or use of the Accused Products without authority or license of Visto.

23. RIM has contributorily infringed and/or induced others to infringe and continues to contributorily infringe and/or to induce others to infringe the '231 patent in this District and elsewhere in the United States by RIM's manufacture, importation, sale, offering for sale, and/or use of the Accused Products without authority or license of Visto.

24. Upon information and belief, RIM's infringement of the '231 patent has been and continues to be willful.

25. RIM's acts have caused, and unless restrained and enjoined, will continue to cause, irreparable injury and damage to Visto for which Visto has no adequate remedy at law. Unless preliminarily and permanently enjoined by this Court, RIM will continue to so infringe the '231 patent.

COUNT IV

(Infringement of U.S. Patent No. 7,228,383)

26. Visto incorporates paragraphs 1 through 25 as though fully restated herein.

27. RIM has infringed and continues to infringe the '383 patent in this District and elsewhere in the United States by RIM's manufacture, importation, sale, offering for sale, and/or use of the Accused Products without authority or license of Visto.

28. RIM has contributorily infringed and/or induced others to infringe and continues to contributorily infringe and/or to induce others to infringe the '383 patent in this District and elsewhere in the United States by RIM's manufacture, importation, sale, offering for sale, and/or use of the Accused Products without authority or license of Visto.

29. Upon information and belief, RIM's infringement of the '383 patent has been and continues to be willful.

30. RIM's acts have caused, and unless restrained and enjoined, will continue to cause, irreparable injury and damage to Visto for which Visto has no adequate remedy at law. Unless preliminarily and permanently enjoined by this Court, RIM will continue to so infringe the '383 patent.

PRAYER FOR RELIEF ON FIRST AMENDED COUNTERCLAIMS

WHEREFORE, Visto respectfully requests this Court to grant it the following relief against RIM on Visto's Counterclaims:

1. A Declaratory Judgment that Visto has not infringed, either directly, contributorily, or by inducement, any valid and enforceable claim of the '839 patent;

2. A Declaratory Judgment that the claims of the '839 patent are invalid;

3. That RIM and its parents, affiliates, subsidiaries, officers, agents, servants, employees, attorneys, successors, and assigns, and all those persons in active concert or participation with them, or any of them, be preliminarily and permanently enjoined and restrained from making, importing, using, offering for sale, selling, or causing to be sold any product falling within, or designed to conduct a method falling within, the scope of any claim of the '231 and '383 Patents, or otherwise infringing or contributing to or inducing infringement of any claim of the '231 and '383 Patents;

4. That RIM and its parents, affiliates, subsidiaries, officers, agents, servants, employees, attorneys, successors, and assigns, and all those persons in active concert or participation with them, or any of them, be ordered to destroy or offer up to Visto for destruction any and all products within the scope of any claim of the '231 and '383 Patents that are within RIM's possession, custody, or control;

5. That Visto be awarded its actual damages, including lost profits and price erosion, but in no case less than a reasonable royalty, to be assessed by or under the Court's discretion, adequate to compensate Visto for RIM's infringement of the '231 and '383 Patents;

6. That Visto be awarded pre-judgment interest and post-judgment interest at the maximum rate allowed by law;

7. That the Court order an accounting for damages;

8. That the Court declare this to be an exceptional case pursuant to 35 U.S.C. § 285 and award Visto its attorneys' fees;

9. That the Court award enhanced damages pursuant to 35 U.S.C. § 284 due to the willful, reckless and wanton nature of RIM's infringement of the '231 and '383 Patents;

10. That Visto be awarded costs of court; and

11. That Visto be awarded such other and further relief as the Court deems just and proper.

Dated: August 29, 2007

MANATT, PHELPS & PHILLIPS, LLP

By: /s/ Robert D. Becker
Robert D. Becker
Eugene L. Hahm
Shawn G. Hansen

Attorneys for Defendant
VISTO CORPORATION

90000198.1

Exhibit 1
to
First Amended Answer to Research in Motion Limited's Complaint for Patent Infringement; Counterclaims Against Research in Motion Limited

US007225231B2

(12) **United States Patent**
Mendez et al.

(10) **Patent No.: US 7,225,231 B2**
(45) **Date of Patent: May 29, 2007**

(54) **SYSTEM AND METHOD FOR TRANSMITTING WORKSPACE ELEMENTS ACROSS A NETWORK**

(75) Inventors: **Daniel J. Mendez**, Mountain View, CA (US); **Mark D. Riggins**, San Jose, CA (US); **Prasad Wagle**, Santa Clara, CA (US); **Christine C. Ying**, Foster City, CA (US)

(73) Assignee: **Visto Corporation**, Redwood Shores, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 289 days.

(21) Appl. No.: **09/921,228**

(22) Filed: **Aug. 1, 2001**

(65) **Prior Publication Data**

US 2002/0035618 A1 Mar. 21, 2002

Related U.S. Application Data

(63) Continuation of application No. 09/666,877, filed on Sep. 20, 2000, now Pat. No. 6,708,221.

(51) **Int. Cl.**
G06F 13/00 (2006.01)

(52) **U.S. Cl.** **709/206**

(58) **Field of Classification Search** 709/248, 709/204, 205, 206, 203; 707/202, 203, 204; 711/161, 162
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,652,698	A	3/1987	Hale et al.
4,714,995	A	12/1987	Materna et al.
4,831,582	A	5/1989	Miller et al.
4,875,159	A	10/1989	Cary et al.
4,882,752	A	11/1989	Lindman et al.
4,897,781	A	1/1990	Chang et al.
4,916,738	A	4/1990	Chandra et al.
5,048,085	A	9/1991	Abraham et al.
5,150,407	A	9/1992	Chan

(Continued)

FOREIGN PATENT DOCUMENTS

CA	CA 2191505	6/1997
CA	CA 2210763	1/1999

(Continued)

OTHER PUBLICATIONS

Emmett Gray, "FETCH-O-MATIC by Emmett Gray", http://www.filmscouts.com/software/fomguide.asp, Version 2.5, 2001.*

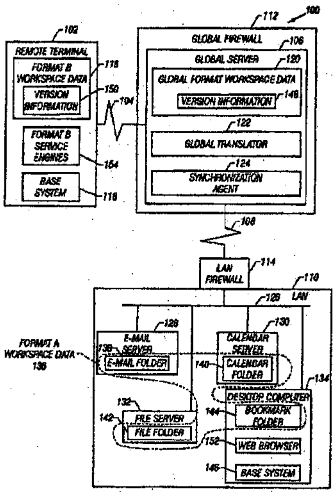
(Continued)

Primary Examiner—Robert B. Harrell
(74) *Attorney, Agent, or Firm*—Oblon, Spivak, McClelland, Maier & Neustadt, P.C.

(57) **ABSTRACT**

A system transmits new workspace elements or new workspace element changes at a first memory store via an electronic network to a second memory store. The system includes a first memory store for storing workspace elements; a second memory store coupled via an electronic network to the first memory store for storing workspace element copies; an interface for receiving new workspace elements at the first memory store; and a general synchronization module for electronically transmitting copies of the new workspace elements via the electronic network to the second memory store.

6 Claims, 5 Drawing Sheets



U.S. PATENT DOCUMENTS

5,220,603	A		6/1993	Parker	
5,263,157	A		11/1993	Janis	
5,265,159	A		11/1993	Kung	
5,333,266	A	*	7/1994	Boaz et al.	709/206
5,386,564	A		1/1995	Shearer et al.	
5,388,255	A		2/1995	Pytlik et al.	
5,392,390	A		2/1995	Crozier	
5,420,927	A		5/1995	Micali	
5,425,102	A		6/1995	Moy	
5,434,918	A		7/1995	Kung et al.	
5,483,596	A		1/1996	Rosenow et al.	
5,491,752	A		2/1996	Kaufman et al.	
5,495,533	A		2/1996	Linehan et al.	
5,510,777	A		4/1996	Pilc et al.	
5,544,320	A		8/1996	Konrad	
5,544,322	A		8/1996	Cheng et al.	
5,572,643	A		11/1996	Judson	
5,581,749	A		12/1996	Hossain et al.	
5,588,132	A		12/1996	Cardoza	
5,600,834	A		2/1997	Howard	
5,604,788	A		2/1997	Tett	
5,613,012	A		3/1997	Hoffman et al.	
5,623,601	A		4/1997	Vu	
5,627,658	A		5/1997	Connors et al.	
5,627,997	A		5/1997	Bray et al.	
5,632,011	A		5/1997	Landfield et al.	
5,634,053	A		5/1997	Noble et al.	
5,644,354	A		7/1997	Thompson et al.	
5,647,002	A		7/1997	Brunson	
5,652,884	A		7/1997	Palevich	
5,657,390	A		8/1997	Elgamal et al.	
5,664,207	A		9/1997	Crumpler et al.	
5,666,530	A	*	9/1997	Clark et al.	707/201
5,666,553	A		9/1997	Crozier	
5,675,782	A		10/1997	Montague et al.	
5,678,039	A		10/1997	Hinks et al.	
5,680,542	A		10/1997	Mulchandani et al.	
5,682,478	A		10/1997	Watson et al.	
5,682,524	A		10/1997	Freund et al.	
5,684,951	A		11/1997	Goldman et al.	
5,684,984	A		11/1997	Jones et al.	
5,684,990	A		11/1997	Boothby	
5,687,322	A		11/1997	Deaton et al.	
5,701,400	A		12/1997	Amado	
5,701,423	A		12/1997	Crozier	
5,706,427	A		1/1998	Tabuki	
5,706,502	A		1/1998	Foley et al.	
5,710,918	A		1/1998	Lagarde et al.	
5,710,922	A		1/1998	Alley et al.	
5,713,019	A		1/1998	Keaten	
5,715,403	A		2/1998	Stefik	
5,717,925	A		2/1998	Harper et al.	
5,721,779	A		2/1998	Funk	
5,721,908	A		2/1998	Lagarde et al.	
5,721,914	A		2/1998	DeVries	
5,727,202	A		3/1998	Kucala	
5,729,735	A		3/1998	Meyering	
5,742,668	A		4/1998	Pepe et al.	
5,745,360	A		4/1998	Leone et al.	
5,752,059	A		5/1998	Holleran et al.	
5,752,246	A		5/1998	Rogers et al.	
5,754,830	A		5/1998	Butts et al.	
5,757,916	A		5/1998	MacDoran et al.	
5,758,150	A		5/1998	Bell et al.	
5,758,354	A		5/1998	Huang et al.	
5,758,355	A		5/1998	Buchanan	
5,764,902	A		6/1998	Rothrock	
5,765,171	A		6/1998	Gehani et al.	
5,768,510	A		6/1998	Gish	
5,778,346	A		7/1998	Frid-Nielsen et al.	
5,778,389	A	*	7/1998	Pruett et al.	707/204
5,784,463	A		7/1998	Chen et al.	
5,784,464	A		7/1998	Akiyama et al.	
5,787,172	A		7/1998	Arnold	
5,790,425	A		8/1998	Wagle	
5,790,790	A		8/1998	Smith et al.	
5,790,974	A		8/1998	Tognazzini	
5,794,252	A		8/1998	Mosher	
5,799,086	A		8/1998	Sudia	
5,799,318	A		8/1998	Cardinal et al.	
5,802,530	A		9/1998	Van Hoff	
5,812,398	A		9/1998	Nielsen	
5,812,668	A		9/1998	Weber	
5,812,773	A		9/1998	Norin	
5,815,683	A		9/1998	Vogler	
5,818,935	A		10/1998	Maa	
5,828,840	A		10/1998	Cowan et al.	
5,832,483	A		11/1998	Barker	
5,835,087	A		11/1998	Herz et al.	
5,835,601	A		11/1998	Shimbo et al.	
5,845,282	A		12/1998	Alley et al.	
5,862,325	A		1/1999	Reed et al.	
5,862,346	A		1/1999	Kley et al.	
5,870,544	A		2/1999	Curtis	
5,870,759	A		2/1999	Bauer et al.	
5,870,765	A		2/1999	Bauer et al.	
5,878,230	A		3/1999	Weber et al.	
5,909,689	A		6/1999	Van Ryzin	
5,924,103	A		7/1999	Ahmed et al.	
5,928,329	A		7/1999	Clark et al.	
5,943,676	A		8/1999	Boothby	
5,951,652	A		9/1999	Ingrassia, Jr. et al.	
5,961,590	A		10/1999	Mendez et al.	
5,966,714	A		10/1999	Huang et al.	
5,968,131	A		10/1999	Mendez et al.	
5,974,238	A	*	10/1999	Chase, Jr.	709/248
5,982,898	A		11/1999	Hsu et al.	
5,987,609	A		11/1999	Hasebe	
5,999,932	A		12/1999	Paul	
5,999,947	A		12/1999	Zollinger et al.	
6,006,017	A		12/1999	Joshi et al.	
6,006,274	A	*	12/1999	Hawkins et al.	709/248
6,020,885	A		2/2000	Honda	
6,021,427	A		2/2000	Spagna et al.	
6,023,700	A		2/2000	Owens et al.	
6,023,708	A		2/2000	Mendez et al.	
6,034,621	A		3/2000	Kaufman	
6,052,735	A		4/2000	Ulrich et al.	
6,073,165	A		6/2000	Narasimhan et al.	
6,085,192	A	*	7/2000	Mendez et al.	707/10
6,094,477	A		7/2000	Nada et al.	
6,125,281	A		7/2000	Wells et al.	
6,108,691	A		8/2000	Lee et al.	
6,108,709	A		8/2000	Shinomura et al.	
6,118,856	A		9/2000	Paarsmarkt et al.	
6,131,096	A		10/2000	Ng et al.	
6,131,116	A		10/2000	Riggins et al.	
6,138,146	A		10/2000	Moon et al.	
6,151,606	A		11/2000	Mendez	
6,154,844	A		11/2000	Touboul et al.	
6,169,986	B1		1/2001	Bowman et al.	
6,182,118	B1		1/2001	Finney et al.	
6,212,529	B1		4/2001	Boothby et al.	
6,249,805	B1		6/2001	Fleming, III	
6,295,541	B1		9/2001	Bodnar et al.	
6,304,881	B1		10/2001	Halim et al.	
6,311,186	B1		10/2001	Melampy et al.	
6,324,542	B1		11/2001	Wright, Jr. et al.	
6,334,140	B1		12/2001	Kawamata	
6,343,313	B1		1/2002	Salesky et al.	
6,389,455	B1		5/2002	Fuisz	
6,438,583	B1		8/2002	McDowell et al.	
6,446,090	B1		9/2002	Hart	
6,477,545	B1		11/2002	LaRue	

6,510,455	B1	1/2003	Chen et al.
6,564,218	B1	5/2003	Roth
6,631,416	B2	10/2003	Bendinelli et al.
6,697,942	B1	2/2004	L'Heureux et al.

FOREIGN PATENT DOCUMENTS

CA	2210763	2/2000
EP	0 801 478	10/1997
EP	EP 0801478	10/1997
EP	EP 0820028	1/1998
JP	JP 02000003314	1/2000
WO	WO 96/17306	6/1996
WO	WO 97/04389	2/1997
WO	WO 97/24678	7/1997
WO	WO 97/35265	9/1997
WO	WO 99/05620	2/1999
WO	WO 99/45451	9/1999
WO	WO 00/22543	4/2000
WO	PCT 2002/0138599	9/2002
WO	PCT 2002/0174372	11/2002

OTHER PUBLICATIONS

Research Disclosure; "Provide Auto-Forwarding Based on Criteria Selected by the User"; Oct. 1, 1989; 1 page; No. 306; Kenneth Mason Publications; XP000085405; ISSN; 0374-4353.

Bellovin, Steven M. et al., "Network Firewalls," IEEE Communications Magazine, Sep. 1994, pp. 50-57.

Braun, Hans-Werner, et al., "Web Traffic Characterization: An Assessment of the Impact of Caching Documents from NCSA's Web Server," Computer Networks and ISDN Systems, 1995, pp. 37-51, vol. 28.

Brown, Margaret J., "The Visto Briefcase Pro Puts Your PIM On The Internet," Aug. 13, 1998, 1 page. URL:http://www.zdnet.com/zdnn/stories/zdnn_display/0,3440,341892,00.html.

Cobb, S., Director of Special Projects National Computer Security Association, "Establishing Firewall Policy," Jun. 25-27, 1996, pp. 198-205, ISBN# 0-7803-3268-7.

Greenwald, Michael, et al., "Designing an Academic Firewall: Policy, Practice, and Experience With SURF," Proceedings of SNDSS 1996, 1996 IEEE, Feb. 22, 1996, pp. 79-92.

Kiuchi, Takahiro, et al., C-HTTP—The Development of a Secure, Closed HTTP-based Network on the Internet, Proceedings of SNDSS 1996, 1996 IEEE, Feb. 22, 1996, pp. 64-75.

Marinacci, Enzo, "Bookmark Translator 2.0—This Utility Transform Microsoft Internet Explore's Bookmarks in the Format Valid for Netscape Nagivator and vice versa," Jul. 1997, 4 pages, Published in Rome, Italy. URL:http://www.bns.it/EMware/Bookmark/Translator-uk.html.

Nelson, Ruth, et al., "Security for Infinite Networks," 1995 IEEE, Aug. 22, 1995, pp. 11-19.

Stempel, Steffen, "IpAccess—An Internet Service Access System for Firewall Installations," 1995 IEEE, Feb. 16, 1995, pp. 31-41.

Radosevich, Linda, "Users want unified mail directories,"Computerworld, Aug. 30, 1993, p.12.

Bruno, Charles, "Firm pushes E-mail limits beyond rivals'," Network World, Aug. 12, 1991, pp. 33, 53.

Sliwa, Carol and Cole, Barb, "MESA declares a groupware detente," Network World, Aug. 5, 1996, p. 29.

Grosse, Eric, "Repository Mirroring," ACM Transactions on Mathematical Software, vol. 21, No. 1, Mar., 1995, pp. 89-97.

Schilit, Bill N. and Theimer, Marvin M., "Disseminating Active map Information to Mobile Hosts," IEEE Network, Sep./Oct. 1994, pp.22-32.

Levy, Eliezer and Silberschatz, Abraham, "Distributed File Systems: Concepts and Examples," ACM Computing Surveys, Vol. 22, No. 4, Dec. 1990, p. 321-74.

Rymer, John R., "The Muddle in the Middle," Byte, Apr. 1996, pp. 67-70.

Baum, David, "Intranet Politics and Technologies," Byte, May 1997, pp. 88A-88H.

Udell, Jon, "Push Me, Pull You," Byte, Sep. 1996, pp. 117-120.

Kador, John, "The Ultimate Middleware," Byte, Apr.1996, pp. 79-83.

Brown, Mark, "Using Netscape 2," Que Corporation, 1995, pp. 885-907.

Kramer, Douglas, "The Java Platform," Javasoft, May 1996, pp. 1-25.

Package java.applet; java.net, http://www-nrg.ee.lbl.gov/j4/apibook/javaf.htm, javam.htm, Apr. 22, 1996.

Servlet Tutorial, http://mech.postech.ac.kr/Java/java.sun.com/products/jeeves/currentrelease/doc/servlet_tutorial/servlet_tutorial.html.

Freier et al., entitled "The SSL Protocol, Version 3.0,"Netscape Communications, dated Nov. 18, 1996, pp. 1-59, Downloaded from URL=http://www.home.netscape.com/eng/ss13/draft302.txt.

Java Servlet Application Programming Interface White Paper, http://mech.postech.ac.kr/Java/java.sun.com/products/jeeves/CurrentRelease/doc/api.html, Aug. 1996, pp. 1-6.

Singhal, M., "Update transport: A new technique for update synchronization in replicated database systems," IEEE Transactions on Software Engineering, vol. 16, No. 12, Dec. 1990, pp. 1325-1336.

Rao, H. & Skarra, A., "A transparent service for syncronized replication across loosely-connected file systems," IEEE Transactions on Software Engineering, Apr. 1995, pp. 110-117.

Crispin, M., "Internet Message Access Protocol - RFC 1730 Version 4," Dec. 1994, pp. 1-52, http://faqs.org/rfcs/rfc1730.html.

Rao, Venkat & Aline, Mary, "Burrowing through firewalls," Dec. 1996, pp.1-5, http://java.sun.com/developer/technicalArticles/InnerWorkings/Burrowing/.

Brown, Margaret J., "The Visto Briefcase Pro Puts Your PIM On The Internet," Aug. 13, 1998, 1 page, URL:http://www.zdnet.com/zdnn/stories/zdnn_display/0,3440,341892,00.html.

Tanenbaum, Andrew, "Computer Networks," Third Edition, Prentice-Hall, 1996, see entire document.

Knudsen, Jonathan, "Java Cryptography," O'Reilly, 1998, pp. 79-91, 112, 160.

Verisign Press Release, "Verisign Enhances Digital IDS to Enable Universal Website Login and One-step Registration," <www.verisign.com/press/product/isv.html>, especially 2nd paragraph.

Crispin, M., "IMAP4 Compatibility With IMAP2BIS", Network Working Group RFC 2061, Dec. 1996, XP002154026, <URL:http://www.imc.org/rfc2061>, retrieved Nov. 28, 2000, page 1, last paragraph.

"Configuration of a wireless-connected mobile computer," IBM Technical Disclosure Bulletin, US, IBM Corp., New York, vol. 38, No. 4, Apr. 1, 1995, pp. 15-17, XP000516057, ISSN: 0018-8689, the whole document.

Mason, Justin, "Tunneling over HTTP," Dec. 11, 1996, pp. 1-2 http://www.netsys.com/firewalls/firewalls-9612/0488.html.

Elgamal, Taher, "The Secure Sockets Layer Protocol (SSL)," agenda for the Danvers IETF Meeting, Apr. 1995, pp. 1-5 - http://www.ietf.cnri.reston.va.us/proceedings/95apr/sec/cat.elgamal.slides.html.

Research Disclosure: "Provide Auto-Forwarding Based On Criteria Selected by the User"; Oct. 1, 1989; 1 page; No. 306; Kenneth Mason Publications; XP000085405; ISSN 0374-4353.

Chan, M.C. et al., "Applications of Compaction Technique to Optimizi ng Wireless Email Transfer," IEEE Wireless Communications and Networking Conference 1999, p. 1535 Section A.

Crispin, M., "Internet Message Access Protocol Version 4rev1," 1996 Section 2.3.1.1.

Jaeger and Prakash, "Implementation of a Discretionary Access Control Model for Script-based Systems," IEEE Jun. 1995.

Adams, Charlotte, "Multilevel Secure Networking Charges Ahead," Federal Computer Week, Apr. 12, 1993.

Kohl, John, "The Evolution of the Kerberos Authentication Service," 1991.

Young et al., "Deniable Password Snatching: On the Possibility of Evasive Electronic Espionage," 1997, IEEE, p. 224-35.

Abel, Amee, "E-mail anywhere," May 1998, Time, pp. 1-4.

Satyanarayanan, M., "Integrating Security in a Large Distributed System," ACM Transactions on Computer Systems, vol. 7, No. 3, Aug. 1989, p. 247-80.

Kistler, James J. and Satyanarayanan, M., "Disconnected Operation in the Coda File System," ACM Transactions on Computer Systems, vol. 10, No. 1, Feb. 1992, pp. 3-25.

Hills, Alex and Johnson, David B., "Wireless Data Network Infrastructure at Carnegie Mellon University," IEEE Personal Communications, 3(1), Feb. 1996.

Satyanarayanan, Mahadev, "Mobile Information Access," IEEE Personal Communications, Feb. 1996, pp. 26-33.

Satyanarayanan, Mahadev et al., "Coda: A Highly Available File System for a Distributed Workstation Environment," IEEE Transactions on Computers, vol. 39, No. 4, Apr. 1990, p. 457-59.

Satyanarayanan, Mahadev, "Scalable, Secure, and Highly Available Distributed File Access," Computer, May 1990, pp. 9-21.

Mummert, Lily B. et al., "Exploiting Weak Connectivity for Mobile File Access," SIGOPS '95, Dec. 1995, p. 143-55.

Terry, Douglas B. et al., "Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System," SIGOPS '95, Dec. 1995, pp. 172-183.

Demers, Alan et al., "The Bayou Architecture: Support for Data Sharing among Mobile Users," Proceedings of the Workshop on Mobile Computing Systems and Applications, Santa Cruz, California, Dec. 1994, pp. 2-7.

Petersen, Karin et al., "Bayou: Replicated Database Services for World-wide Applications," Proceedings Seventh ACM SIGOPS European Workshop (EuroSIGOPS '96), Connemara, Ireland, Sep. 1996, pp. 275-280.

Crocker, David H., "RFC822: Standard for ARPA Internet Text Messages," http://www.w3.org/Protocols/rfc822/.

Theisen, Tim, "AFS distributed filesystem FAQ (1/2)," posted to uwisc.general newsgroup, Jul. 25, 1994.

Theisen, Tim, "AFS distributed filesystem FAQ (2/2)," posted to uwisc.general newsgroup, Jul. 25, 1994.

Schubert, Eric, "Re: telnet/internet and ... ," posted to comp.sys.hp.mpe newsgroup, Jun. 30, 1995.

Glazman, Daniel, "SOFTWARE:HHTPtool [sic] 1.1, a file transfer utility over HTTP using PUT and GET," posted to comp.infosystems.www.announce newsgroup, May 17, 1995.

Glazman, Daniel, "SOFTWARE:HHTPtool v1.0 for Windows3.x, file transfer utility over HTTP," posted to comp.infosystems.www.announce newsgroup, Mar. 21, 1996.

Angus, Jeffrey G., "Sales force automation has a GoldMine," Computerworld, Oct. 7, 1996, p. 59.

Salamone, salvatore, "Middle(ware) Management," Byte, Apr. 1996, pp. 71-76.

Nance, Barry, "Balance the Load with Transaction Server," Byte, Jun. 1997, pp. 81-84.

Francett, barbara, "Replication on the Run," Software Magazine, Aug. 1996, pp. 63-66.

Darling, Charles B., EDA/SQL Loses a Little and Gains a Lot, Datamation, May 1, 1996, p. 12.

Varney, Sarah E., "Arm your salesforce with the Web," Datamation, Oct. 1996, pp. 72-74.

Fulcher, Jim, "Is it or isn't it?," Manufacturing Systems, Oct. 1996, pp. 56-61.

Lamb, John and Cusato, Tony, "LAN-Based Office for the Enterprise, A Case Study," Proceedings, 19th Conference on Local Computer Networks, Minneapolis, Minnesota, Oct. 2-5, 1994, pp. 440-447.

Kawell Jr., Leonard et al., "Replicated Document Management In A Group Communication System," presented at the Second Conference on Computer-Supported Cooperative Work, Portland, Oregon, Sep. 26-28, 1988, as printed in Groupware: Software for Computer-Supported Cooperative Work, IEEE Computer Society Press, pp. 226-235.

Hong, Jack et al., "Personal Electronic Notebook with Sharing," Proceedings of the Fourth Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, Berkley Springs, West Virginia, Apr. 20-22, 1995, pp. 88-94.

Mace, Scott, "DataSync 2.0 enhances synchronization of data," InfoWorld, Jun. 6, 1994, p.28.

Mace, Scott, "Database replication products debut, " InfoWorld, Nov. 15, 1993, p.19.

Gryphon, Robert, "DataSync line matches up," InfoWorld, May 23, 1994, p. 161.

Patterson, John F. et al., "Notification Servers for Synchronous Groupware," Proceedings, ACM 1996 Conference on Computer Supported Cooperative Work, Boston, Massachusetts, Nov. 16-20, 1996, pp. 122-129.

Zhao, Jian and Hoppe, H. Ulrich, "Supporting Flexible Communication in Heterogeneous Multi-User Environments," 1994 IEEE, pp. 442-449.

Gong, Li, "Enclaves: Enabling Secure Collaboration Over the Internet," IEEE Journal on Selected Areas in Communications, vol. 15, No. 3, Apr. 1997, pp. 567-575.

Bruno, Lee, "Directory Services Tie It All Together," Data Communications, Mar. 1997, pp. 75-83.

Guy, Richard G. et al., "Implementation of the Ficus Replicated File System," Proceedings of the Summer 1990 USENIX Conference, Anaheim, California, Jun. 11-15, 1990, pp. 63-70.

Page, Thomas W. Jr. et al., "Management of Replicated Volume Location Data in the Ficus Replicated Data System," Conference Proceedings, USENIX Summer 1991 Technical Conference, Nashville, Tennessee, Jun. 10-14, 1991, pp. 17-29.

Reiher, Peter et al., "Resolving File Conflicts in the Ficus File System," Conference Proceedings, USENIX Summer 1994 Technical Conference, Boston, Massachusetts, Jun. 6-10, 1994, pp. 183-195.

Walker, Bruce et al., "The LOCUS Distributed Operating System," Proceedings of the Ninth ACM Symposium on Operating Systems Principles, Bretton Woods, New Hampshire, Oct. 10-13, 1983, pp. 49-70.

Parker, D. Stott Jr. et al., "Detection of Mutual Inconsistency in Distributed Systems," IEEE Transactions on Software Engineering, vol. SE-9, No. 3, May 1983, pp. 240-246.

Ceri, Stefano et al., "The Case for Independent Updates", Second Workshop on the Management of Replicated Data, Monterey, California, Nov. 12-13, 1992, pp. 17-19.

Downing, Alan R. et al., "OSCAR: A System for Weak-Consistency Replication", Proceedings, Workshop on the Management of Replicated Data, Houston, Texas, Nov. 8-9, 1990, pp. 26-30.

Siegel, Alex et al., "Deceit: A Flexible Distributed File System", Proceedings of the Summer 1990 USENIX Conference, Anaheim, California, Jun. 11-15, 1990, pp. 51-61.

Chutani, Sailesh, "The Episode File System", Conference Proceedings, USENIX Winter 1992 Technical Conference, San Francisco, California, Jan. 20-24, 1992, pp. 43-59.

Seltzer, Margo, "An Implementation of a Log-Structured File System for UNIX", Conference Proceedings, USENIX Winter 1993 Technical Conference, San Diego, California, Jan. 25-29, 1993, pp. 307-326.

Vahalia, Uresh, "Metadata Logging in an NFS Server", Conference Proceedings, USENIX 1995 Technical Conference on UNIX and Advanced Computing Systems, New Orleans, Louisiana, Jan. 16-20, 1995, pp. 265-276.

Lotus Development Corporation, InterNotes Web Publisher Release 4, "InterNotes Web Publisher Guide", 1996, Cambridge, MA.

Lotus Development Corporation, Lotus Notes Release 4, "Database Manager's Guide", 1995, Cambridge, MA.

Lotus Development Corporation, Lotus Notes Release 4, "Administrator's Guide", 1995, Cambridge, MA.

IBM Lotus Technical Library, Lotus Documentation, "Lotus Notes Internet Cookbook for Notes Release 3", Jan. 16, 1996, pp. 1-26.

John Wiley & Sons, Inc. — Wiley Computer Publishing, Falkner, Mike, "How to Plan, develop, and Implement Lotus Notes in Your Organization", 1996, USA4.

Lotus Development Corporation, Lotus Notes Release 4, "Deployment Guide", 1995, Cambridge, MA.

McGraw-Hill, Lamb, John P. and Lew, Peter W., "Lotus Notes Network Design for Notes Release 3 and 4", 1996, Quebecor-Fairfield, PA.

Lotus Notes, "Overview — What is Lotus NotesPump?", including "notes Pump 1.0 Release Notes."

Lotus Development Corporation, Lotus Notes Release 3.1, The groupware standard, "Site and System Planning Guide", 1994, Cambridge, MA.

Lotus Development Corporation, Lotus Notes Release 3.1. The groupwarestandard, "Administrator's Guide Server for NetWare, OS/2, and UNIX", 1994, Cambridge, MA.

IBM Lotus Technical Library, Lotus Documentation, "Lotus Notes Internet Cookbook for Notes Release 4", Feb. 14, 1996, pp. 1-30 http://www-12.lotus.com/ldd/doc/domino_notes/cookbook/cbookv4,nsf/c125003289bf7b3a385256.

IntelliLink Corporation, IntelliLink for Windows Release 3.0, "User's Guide", 1994, Nashua, NH.

Lotus Development Corporation, Lotus Notes Release 4, "Application Developer's Guide", 1995, Cambridge, MA.

Lotus Development Corporation, Lotus Notes Release 3.3 North American Server Edition, "Lotus Notes, the Groupware Standard", 1994, Cambridge, MA.

Sams Publishing, Dahl, Andrew, "Lotus Notes 4 Administrator's Survival Guide", 1996, Indianapolis, IN.

Sams Publishing, Tmaura, et al., "Lotus Notes 4 Unleashed", 1996, Indianapolis, IN.

Advisor Publications — Lotus Notes Advisor, Pyle, Hugh, "The Notes Architecture", 1995.

Advisor Publications — Lotus Notes Advisor, Augun, Adury, "Integrating Lotus Notes with Enterprise Data", 1996.

Advisor Publications — Lotus Notes Advisor, Opyt, Barbara and Dale, Robert, "Use the Internet as Your Lotus Notes WAN", 1996.

Lotus Development Corporation, Lotus Notes Knowledge Base, "What is the Notes Replicator?", 1995, Cambridge, MA.

Lotus Development Corporation, Lotus Notes Knowledge Base, "Firewall Security Overview and How Firewalls Relate to Lotus Notes", 1996, Cambridge, MA.

Network Computing, Frenkel, Garry, "Pumping for Info: Notes and Database Integration", 1996.

IBM Corporation, Hawker, et al., "Secrets to Running Lotus Notes: The Decisions No One Tells You How to Make", 1996, Research Triangle Park, NC.

US 5,373,559, 12/1994, Kaufman et al. (withdrawn)

* cited by examiner

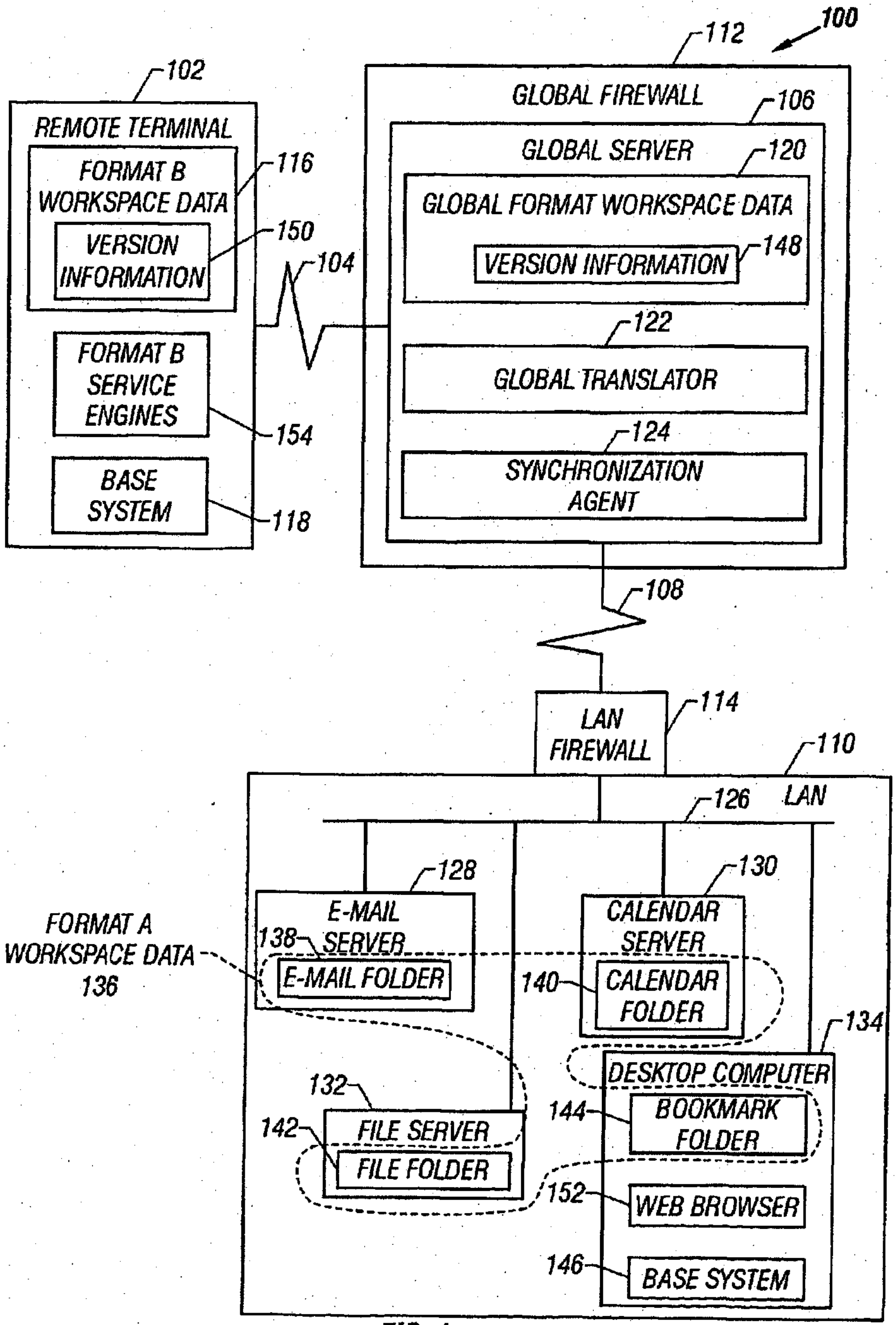

100
102
REMOTE TERMINAL
FORMAT B WORKSPACE DATA
116
VERSION INFORMATION
150
104
FORMAT B SERVICE ENGINES
154
BASE SYSTEM
118
112
GLOBAL FIREWALL
106
GLOBAL SERVER
120
GLOBAL FORMAT WORKSPACE DATA
VERSION INFORMATION
148
122
GLOBAL TRANSLATOR
124
SYNCHRONIZATION AGENT
108
LAN FIREWALL
114
110
LAN
126
128
E-MAIL SERVER
138
E-MAIL FOLDER
130
CALENDAR SERVER
140
CALENDAR FOLDER
FORMAT A WORKSPACE DATA 136
134
DESKTOP COMPUTER
144
BOOKMARK FOLDER
132
FILE SERVER
142
FILE FOLDER
152
WEB BROWSER
146
BASE SYSTEM

FIG. 1

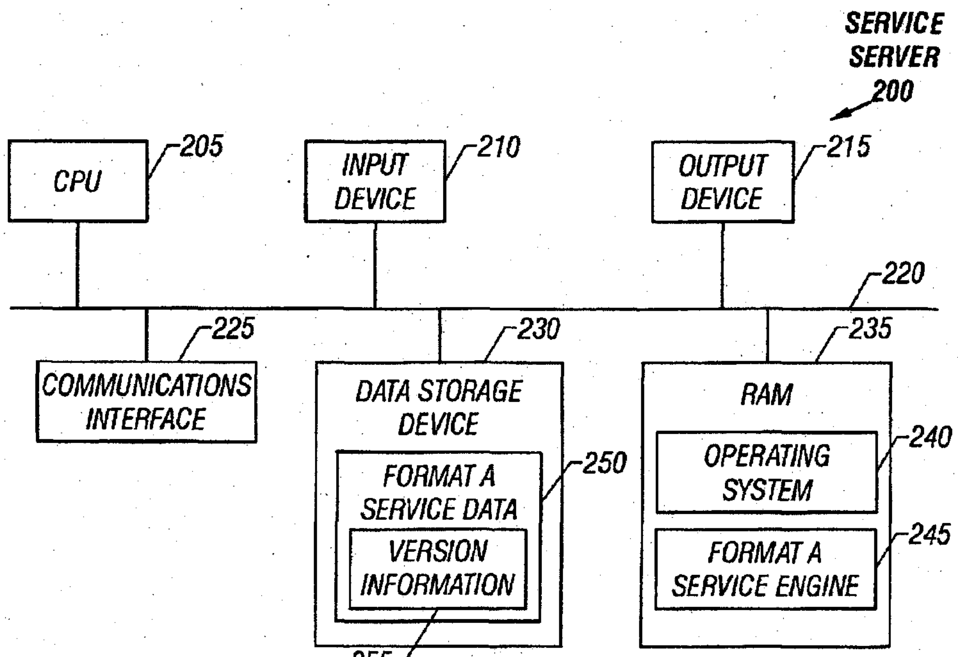

SERVICE SERVER 200
CPU
205
INPUT DEVICE
210
OUTPUT DEVICE
215
220
COMMUNICATIONS INTERFACE
225
DATA STORAGE DEVICE
230
RAM
235
OPERATING SYSTEM
240
FORMAT A SERVICE DATA
250
VERSION INFORMATION
255
FORMAT A SERVICE ENGINE
245

FIG. 2

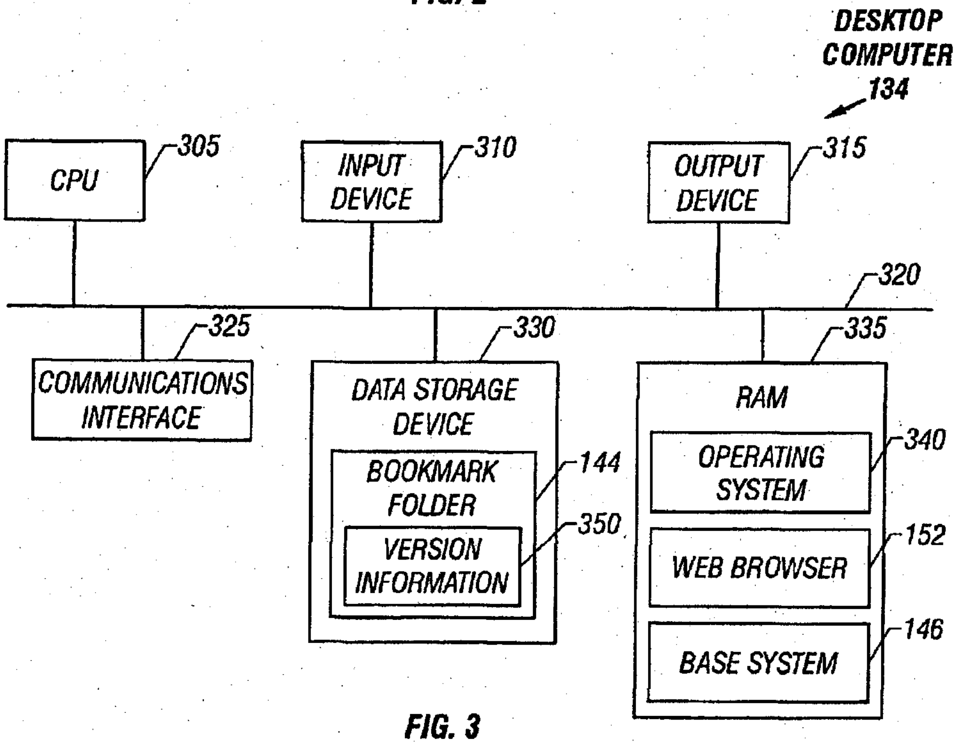

DESKTOP COMPUTER 134
CPU
305
INPUT DEVICE
310
OUTPUT DEVICE
315
320
COMMUNICATIONS INTERFACE
325
DATA STORAGE DEVICE
330
RAM
335
OPERATING SYSTEM
340
BOOKMARK FOLDER
144
VERSION INFORMATION
350
WEB BROWSER
152
BASE SYSTEM
146

FIG. 3

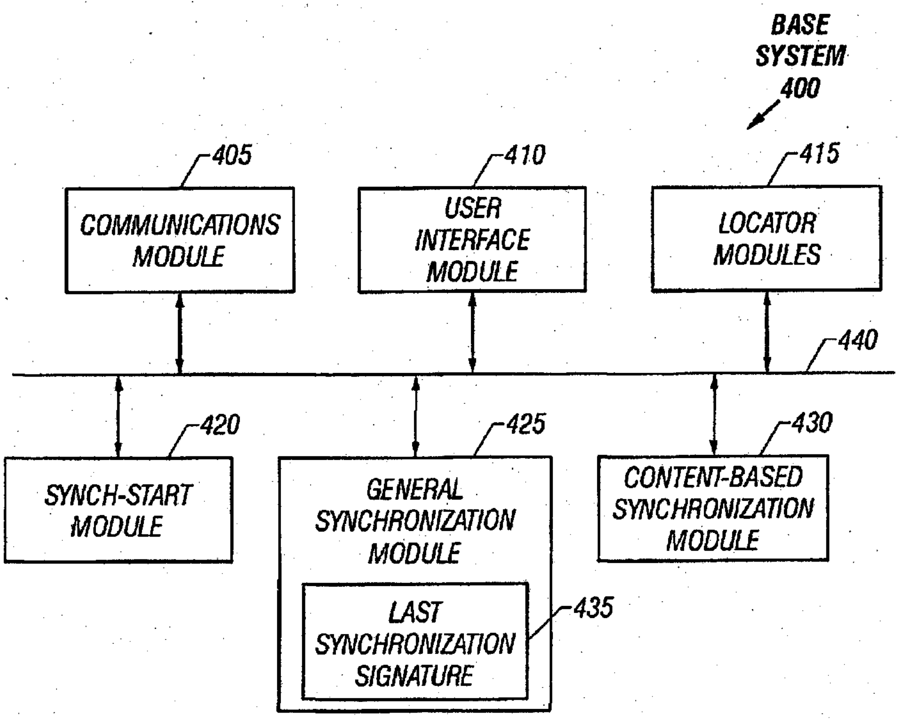

BASE SYSTEM 400
405
COMMUNICATIONS MODULE
410
USER INTERFACE MODULE
415
LOCATOR MODULES
440
420
SYNCH-START MODULE
425
GENERAL SYNCHRONIZATION MODULE
LAST SYNCHRONIZATION SIGNATURE
435
430
CONTENT-BASED SYNCHRONIZATION MODULE

FIG. 4

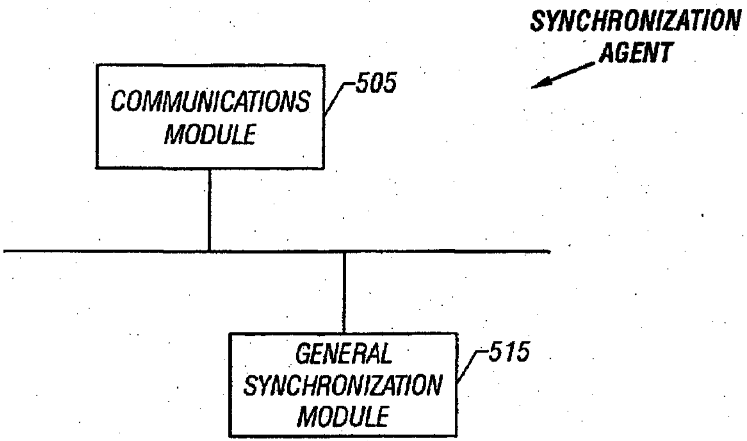

SYNCHRONIZATION AGENT
COMMUNICATIONS MODULE
505
GENERAL SYNCHRONIZATION MODULE
515

FIG. 5

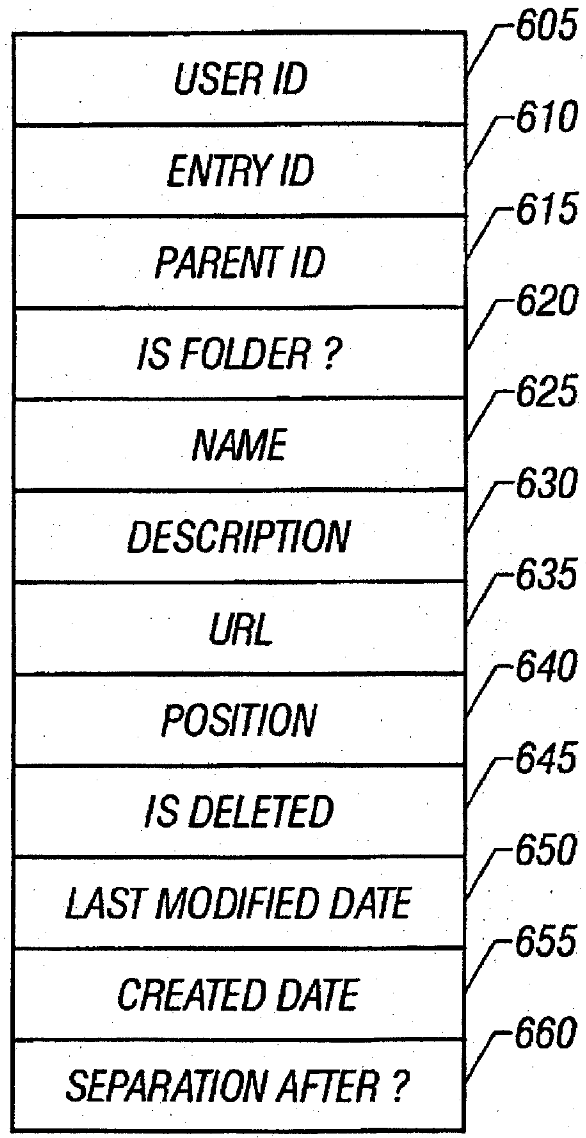

USER ID
605
ENTRY ID
610
PARENT ID
615
IS FOLDER ?
620
NAME
625
DESCRIPTION
630
URL
635
POSITION
640
IS DELETED
645
LAST MODIFIED DATE
650
CREATED DATE
655
SEPARATION AFTER ?
660

FIG. 6

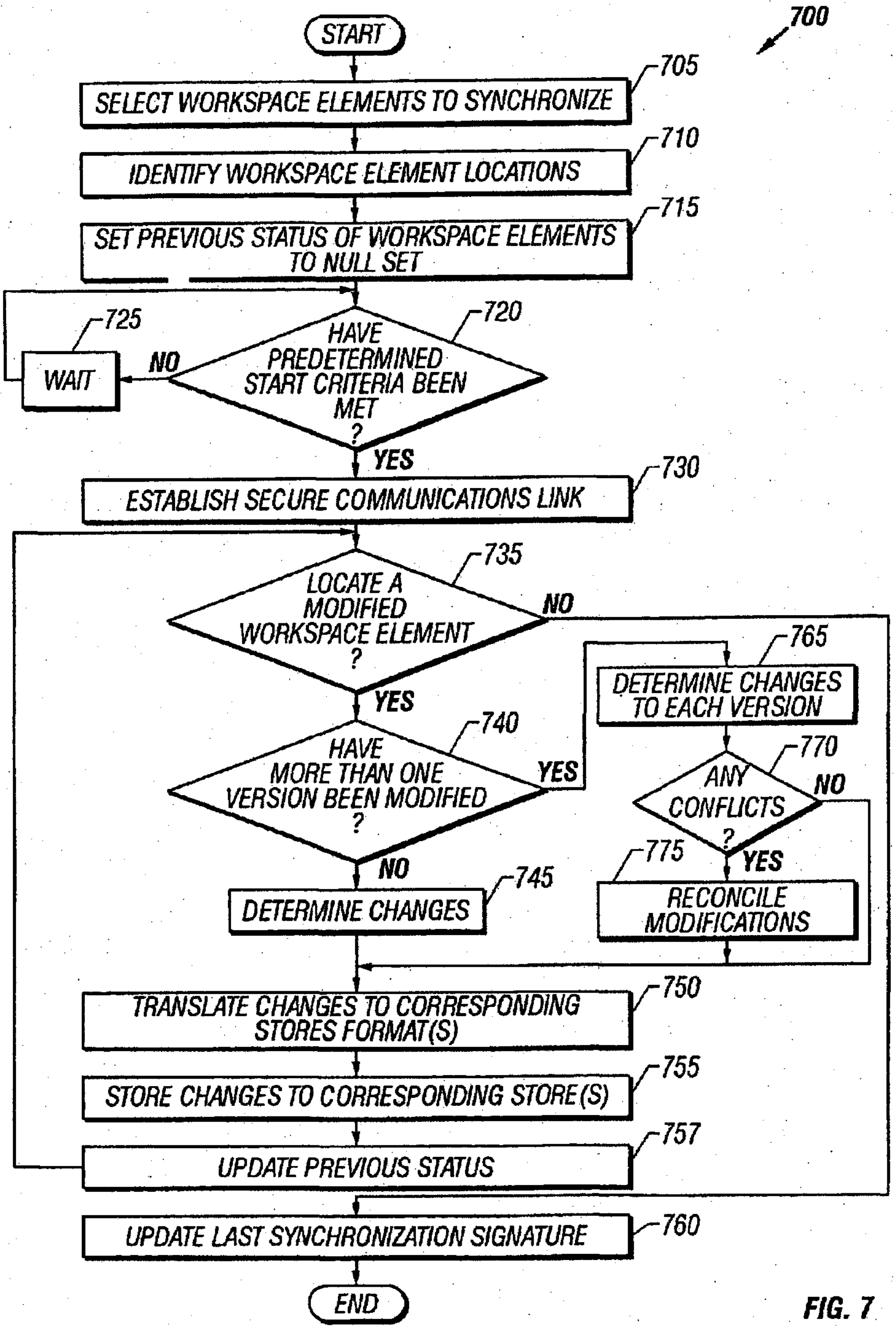

700
START
SELECT WORKSPACE ELEMENTS TO SYNCHRONIZE
705
IDENTIFY WORKSPACE ELEMENT LOCATIONS
710
SET PREVIOUS STATUS OF WORKSPACE ELEMENTS TO NULL SET
715
HAVE PREDETERMINED START CRITERIA BEEN MET ?
720
NO
WAIT
725
YES
ESTABLISH SECURE COMMUNICATIONS LINK
730
LOCATE A MODIFIED WORKSPACE ELEMENT ?
735
NO
YES
HAVE MORE THAN ONE VERSION BEEN MODIFIED ?
740
YES
NO
DETERMINE CHANGES TO EACH VERSION
765
ANY CONFLICTS ?
770
NO
YES
775
RECONCILE MODIFICATIONS
DETERMINE CHANGES
745
TRANSLATE CHANGES TO CORRESPONDING STORES FORMAT(S)
750
STORE CHANGES TO CORRESPONDING STORE(S)
755
UPDATE PREVIOUS STATUS
757
UPDATE LAST SYNCHRONIZATION SIGNATURE
760
END

FIG. 7

SYSTEM AND METHOD FOR TRANSMITTING WORKSPACE ELEMENTS ACROSS A NETWORK

CROSS-REFERENCE TO PRIORITY APPLICATION

This application is a continuation claims priority to the patent application entitled "System and Method for Globally and Securely Accessing Unified Information in a Computer Network," Ser. No. 09/666,877, filed on Sep. 20, 2000, now U.S. Pat. No. 6,708,221 by inventors Daniel J. Mendez, Mark D. Riggins, Prasad Wagle, Hong Q. Bui, Mason Ng, Sean Micheal Quinlan, Christine C. Ying, Christopher R. Zuleeg, David J. Cowan, Joanna A. Aptekar-Strober and R. Stanley Bailes, and hereby incorporates by reference U.S. Pat. No. 6,085,192, issued on Jul. 4, 2000, entitled "System and Method for Securely Synchronizing Multiple Copies of a Workspace Element in a Network," application Ser. No. 08/835,997, filed on Apr. 11, 1997, by inventors Daniel J. Mendez, Mark D. Riggins, Prasad Wagle and Christine C. Ying.

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates generally to computer networks, and more particularly to a system and method for using a global translator to synchronize workspace elements such as files across a computer network.

2. Description of the Background Art

Data consistency is a significant concern for computer users. For example, when maintaining multiple independently-modifiable copies of a document, a user risks using an outdated version. By the time the user notices an inconsistency, interparty miscommunication or data loss may have already resulted. The user must then spend more time attempting to reconcile the inconsistent versions and addressing any miscommunications.

The problem of data inconsistency is exacerbated when multiple copies of a document are maintained at different network locations. For example, due to network security systems such as conventional firewall technology, a user may have access only to a particular one of these network locations. Without access to the other sites, the user cannot confirm that the version on the accessible site is the most recent draft.

Data consistency problems may also arise when using application programs from different vendors. For example, the Netscape Navigator™ web browser and the Internet Explorer™ web browser each store bookmarks for quick reference to interesting web sites. However, since each web browser uses different formats and stores bookmarks in different files, the bookmarks are not interchangeable. In addition, one web browser may store a needed bookmark, and the other may not. A user who, for example, runs the Internet Explorer™ web browser at home and runs the Netscape Navigator™ web browser at work risks having inconsistent bookmarks at each location.

Therefore, a system and method are needed for providing users with data consistency, and more particularly for synchronizing multiple copies of a workspace element such as a document across a computer network.

SUMMARY OF THE INVENTION

The present invention provides a system and method for using a global translator to synchronize multiple copies of a workspace element in a secure network environment. The secure network environment includes a global server connected to multiple clients. Using the present system and method, the clients automatically synchronize workspace elements between multiple sites, independent of whether the sites are protected by site firewalls. Using the present system and method, the clients can automatically synchronize workspace elements across different formats and can merge workspace element folders for cross use.

The system includes a first store for storing first workspace elements in a first format, a second store for storing second workspace elements in a second format, a communications channel coupling the first store to the second store, synchronization means for synchronizing first workspace elements and second workspace elements, and a translator for translating between the first format and the second format.

Similarly, the method includes the steps of accessing a first store storing a first workspace element in a first format, accessing a second store storing a second workspace element in a second format, synchronizing the first workspace element and the second workspace element, and translating between the first format and the second format.

The system and method advantageously use a trusted third party to enable the synchronization of workspace data among multiple sites. Accordingly, a client user who maintains a work site, a home site, an off-site and the global server site can synchronize the workspace data or portions thereof among all four sites. Further, the predetermined criteria (which control when the synchronization-start module initiates synchronization) may be set so that the general synchronization module synchronizes the workspace data upon user request, at predetermined times during the day such as while the user is commuting, or after a predetermined user action such as user log-off or user log-on. Because the system and method operate over the Internet, synchronization can occur over any distance. Since the system and method include format translation, merging of workspace elements between different application programs and different platforms is possible. Further, because synchronization is initiated from within the firewall, the typical firewall which prevents in-bound communications does not act as an impediment to workspace element synchronization. Also, since the user's preferences may be previously set, the present system and method may operate unattended by the client user.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram illustrating a computer network in accordance with the present invention;

FIG. **2** is a block diagram illustrating details of a FIG. **1** service server;

FIG. **3** is a block diagram illustrating details of the FIG. 1 desktop computer;

FIG. **4** is a block diagram illustrating details of a FIG. 1 base system;

FIG. **5** is a block diagram illustrating details of the FIG. 1 synchronization agent;

FIG. **6** is a graphical representation of an example bookmark in the global format; and

FIG. **7** is a flowchart illustrating a method for synchronizing multiple copies of a workspace element in a secure network.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. **1** is a block diagram illustrating a computer network **100**, comprising a first node such as a remote computer terminal **102** coupled via a communications channel **104** such as the Internet to a global server **106**. The global server **106** is in turn coupled via a communications channel **108** such as the Internet to a second node such as a Local Area Network (LAN) **110**. The global server **106** is protected by a global firewall **112**, and the LAN **110** is protected by a LAN firewall **114**.

The LAN **110** includes a system bus **126** coupling the LAN firewall **114** to an e-mail server **128** having an e-mail folder **138** containing e-mails, to a file server **132** having a file folder **142** containing files, to a calendar server **130** having a calendar folder **140** containing calendar data, and to a desktop computer **134** having a web browser **152** and a bookmark folder **144** containing bookmarks. It will be appreciated that the e-mail folder **138**, file folder **142**, calendar folder **140** and bookmark folder **144** or portions thereof may be stored at different locations such as on the desktop computer **134**. The e-mail folder **138**, file folder **142**, calendar folder **140** and bookmark folder **144** are exemplary, grouped by like information and are collectively referred to herein as "workspace data" **136**. Those skilled in the art will recognize that the workspace data **136** may include other types of data such as an application program such as Microsoft Word 6.0.1 and the documents created using them. It will be further appreciated that the e-mail folder **138**, file folder **142**, calendar folder **140** and bookmark folder **144** may each be divided into workspace elements, wherein each workspace element folder or each workspace element individually is identified by particular version information **255** (described below with reference to FIG. **2**). Accordingly, each e-mail or e-mail folder, file or file folder, calendar or calendar folder, bookmark or bookmark folder, document or document folder, etc. may be referred to as "a workspace element."

Each workspace element of workspace data **136** in LAN **110** is maintained in a predetermined format, referred to as Format A, which is based on the service engine **245** (FIG. **2**) that created it. For example, the web browser **152** on the desktop computer **134** may be the Netscape Navigator™ web browser, and the bookmarks in the bookmark folder **144** created thereby are maintained in Format A. Although Format A is being described as a single format, one skilled in the art knows that Format A actually includes a format for each information type, e.g., there will be a Format A for bookmarks, a Format A for files, a Format A for calendar data, a Format A for e-mails, etc.

The remote terminal **102** stores service engines **154** for maintaining workspace data **116**, which may include information common with information in the workspace data **136**. The workspace data **116** is maintained in a format, referred to as Format B, which may be different from Format A. Format B is also based on the service engines **154** that create the workspace elements. For example, if one of the service engines **154** is the Internet Explorer™ web browser (not shown), then the bookmarks (not shown) created therewith are maintained in Format B. Although Format B is being described as a single format, one skilled in the art knows that Format B actually includes a format for each information type. Further, the workspace data **116** also includes version information **150** similar to version information **255** described below with reference to FIG. **2**.

It will be appreciated that remote terminal **102** may include a smart telephone, a Personal Data Assistant (PDA) such as the PalmPilot system by the U.S. Robotics, Inc., a laptop computer, etc. As a smart telephone, the workspace data **116** may include telephone numbers and e-mails. As a PDA, the workspace data **116** may include addresses, calendar data and e-mails. As a laptop computer, the workspace data **116** may include the same types of information as workspace data **136**.

The global server **106** acts as a third party administrator. The global server **106** stores independently-modifiable copies of selected portions of the workspace data **136** and **116**, collectively referred to herein as workspace data **120**. Accordingly, the workspace data **120** includes an independently-modifiable copy of each workspace element in the selected portions of the workspace data **136** and **116** and an independently-modifiable copy of each corresponding version information **255** (FIG. **2**) and **150**. The version information copies are collectively referred to herein as version information **148**, and are also described with reference to FIG. **2**.

The global server **106** maintains the workspace data **120** in a format, referred to as a "global format," which is selected to be easily translatable by the global translator **122** to and from Format A and to and from Format B. Although the global format is being described as a single format, one skilled in the art knows that the global format actually includes a global format for each information type, e.g., there will be a global format for bookmarks, a global format for files, a global format for calendar data, a global format for e-mails, etc. An example bookmark workspace element in the global format is described in detail below with reference to FIG. **6**.

Network **100** further comprises synchronization means, which includes a base system **146** stored within the LAN **110** and for example on the desktop computer **134**. Network **100** further includes a synchronization agent **124** stored outside the LAN firewall **114** and preferably on the global server **106**. The base system **146** and the synchronization agent **124** cooperate to synchronize selected portions of the workspace data **136** with selected portions of the workspace data **120**. The synchronization means may synchronize workspace elements individually, e.g., specific word processor documents, or may synchronize workspace element folders, e.g., a bookmark folder. Generally, the base system **146** manages the selected portion of the workspace data **136** within the LAN **110** and the synchronization agent **124** manages the selected portions of the workspace data **120** within the global server **106**. It will be appreciated that the global translator **122** cooperates with the synchronization means to translate data formats to and from the global format. As described in greater detail below with reference to FIG. **4**, the base system **190** preferably initiates and controls data synchronization. Other components and functions of the global server **106** are described in the cross-referenced patent application which is herein incorporated by reference.

The synchronization means may also include, stored on the remote terminal **102**, a base system **118** which operates in a similar manner to the base system **146**. The base system **118** on the remote terminal **102** cooperates with the synchronization agent **124** to synchronize selected portions of the workspace data **116** with selected portions of the workspace data **120**. As described in greater detail below with reference to FIG. **4**, the base system **118** on the remote terminal **102** also preferably initiates and controls data synchronization with the global server **106**. Also, note that the distribution of labor between the base system **118** in the remote terminal **102** and the synchronization agent **124** in the global server **106** may vary. Sometimes, primarily when the remote terminal **102** is a relatively less computationally powerful device (such as a smart phone or a PDA), most of the actual computationally-intensive work will occur within the synchronization agent **124** in the global server **106**. In other situations, for example, when the remote terminal **102** is a fully configured PC, most of the computationally-intensive work will occur locally on the base system **118** in the remote terminal **102**.

Accordingly, the synchronization means independently synchronizes the selected portions of workspace data **116** and **136** with the selected portions of the workspace data **120**. Thus, the synchronization means indirectly synchronizes workspace data **136** with workspace data **116**.

FIG. **2** is a block diagram illustrating details of a service server **200**, wherein each of the e-mail server **145**, the file server **150**, the calendar server **155** and the desktop computer **160** is an instance thereof. Service server **200** includes a Central Processing Unit (CPU) **205** such as an Intel Pentium® microprocessor or a Motorola Power PC® microprocessor. An input device **210** such as a keyboard and mouse and an output device **215** such as a Cathode Ray Tube (CRT) display are coupled via a signal bus **220** to CPU **205**. A communications interface **225** (such as an Ethernet port), a data storage device **230** (such as a magnetic disk), and Random-Access Memory (RAM) **235** are further coupled via signal bus **220** to the CPU **205**.

An operating system **240** includes a program for controlling processing by the CPU **205**, and is typically stored in the data storage device **230** and loaded into the RAM **235** for execution. A service engine **245** includes a program for performing a particular service such as maintaining an e-mail data base, a file data base, a calendar data base or a bookmarks data base. The service engine **245** may also be stored in the data storage device **230** and loaded into the RAM **235** for execution.

To perform a service, the service engine **245** creates service data **250** (e.g., an e-mail or an e-mail folder **138** containing e-mails, a file or a file folder **142** containing files, calendar data or a calendar folder **140** containing calendar data, a bookmark or a bookmark folder **144** containing bookmarks, etc.) in Format A according to predetermined protocols. The service engine **245** stores the data **250** in the data storage device **250**. The service data **250** includes version information **255** indicating the date and time of the last modification and the status as of the last interaction with the global server **106**.

For example, if service data **250** is created and selected to be merged with global server workspace data **120**, then the version information **255** for the service data **250** may include the date of last modification and a null set indicating the status as of the last interaction with the global server **106**. From the version information **255**, the base system **146** determines that the service data **250** in its entirety has not been merged with the global server workspace data **120**. Similarly, if the service data **255** included elements **1**, **2** and **3** as of the last modification, then the previous status as of the last interaction will indicate that the service data **255** included elements **1**, **2** and **3**. If the service data **255** currently includes elements **2**, **3** and **4**, then the base system **140** will determine, that, since last synchronization, element **1** has been deleted and element **4** has been added.

It will be appreciated that the version information **148** on the global server **106** includes information similar to version information **255**. That is, the version information **148** will include information indicating the date and time the version was last modified and the status as of the last interaction with each client. The service engine **245** operates to update the version information **255** after modifications are made and after synchronization occurs.

FIG. **3** is a block diagram illustrating details of the desktop computer **160**, which includes a CPU **305**, an input device **310**, an output device **315**, a communications interface **325**, a data storage device **330** and RAM **335**, each coupled to a signal bus **320**.

An operating system **340** includes a program for controlling processing by the CPU **305**, and is typically stored in the data storage device **330** and loaded into the RAM **335** for execution. A web browser **152** (i.e., a particular service engine **245**, FIG. **2**) includes a Format A service program for managing bookmark folder **144** (i.e., particular service data **250**, FIG. **2**) which includes version information **350** (i.e., particular version information **255**, FIG. **2**). The web browser **152** may be also stored in the data storage device **330** and loaded into the RAM **335** for execution. The bookmark folder **144** may be stored in the data storage device **330**. As stated above with reference to FIG. **1**, the base system **146** operates to synchronize the workspace data **136** (which includes the bookmark folder **144**) with the workspace data **120**. The base system **146** may be also stored in the data storage device **330** and loaded into the RAM **335** for execution.

FIG. **4** is a block diagram illustrating details of the base system **400**, which exemplifies base systems **146** and **118**. Base system **400** includes a communications module **405**, a user interface module **410**, locator modules **415**, a synchronization-start ("synch-start") module **420**, a general synchronization module **425** and a content-based synchronization module **430**. For simplicity, each module is illustrated as communicating with one another via a signal bus **440**.

The communications module **405** includes routines for compressing data and routines for communicating via the communications interface **325** (FIG. **3**) with the synchronization agent **124** (FIG. **1**). The communications module **405** may further include routines for applying Secure Socket Layer (SSL) technology and user identification and authentication techniques (i.e., digital certificates) to establish a secure communication channel through the global firewall **112**. Examples of communications modules **405** may include TCP/IP stacks or the AppleTalk® protocol.

The user interface module **410** includes routines for communicating with a user, and may include a conventional Graphical User Interface (GUI). The user interface module **410** cooperates with the other system components as described herein.

The locator modules **415** include routines for identifying the memory locations of the workspace elements in the workspace data **136** or **116** and in the workspace data **120**. Workspace element memory location identification may be implemented using intelligent software, i.e., preset memory addresses or the system's registry, or using dialogue boxes to query a user. More particularly, the locator modules **415** in the base system **146** determine the memory addresses of the e-mail folder **138**, the file folder **142**, the calendar folder **140** and the bookmark folder **144** and the memory addresses of the workspace elements therein. The locator modules **415** also determine the corresponding memory addresses of the corresponding folders in the workspace data **120** and the corresponding workspace elements therein. Similarly, the locator modules **415** in the base system **118** determine the memory locations of the workspace elements of workspace data **116** and the memory locations of the corresponding workspace elements in the workspace data **120**.

It will be appreciated that the locator modules **415** may include locator modules **415** specifically dedicated to each folder or workspace data type. That is, the locator modules **415** may include a locator module **415** dedicated to locating bookmarks, a locator module **415** dedicated to locating e-mails, a locator module **415** dedicated to locating files, a locator module **415** dedicated to locating calendar appointments, etc. It will be further appreciated that the locator modules **415** may perform workspace element memory location identification upon system boot-up or after each communication with the global server **120** to maintain updated memory addresses of workspace elements.

The synchronization-start module **420** includes routines for determining when to initiate synchronization of workspace data **136** or **116** with workspace data **120**. For example, the synchronization-start module **420** may initiate data synchronization upon user request, at a particular time of day, after a predetermined time period passes, after a predetermined number of changes, after a user action such as user log-off or upon like criteria. The synchronization-start module **420** initiates data synchronization by instructing the general synchronization module **425** (described below) to begin execution of its routines. It will be appreciated that communication with the synchronization agent **124** preferably initiates from within the LAN **110**, because the typical firewall **114** prevents in-bound communications and allows out-bound communications.

The general synchronization module **425** includes routines for receiving version information **148** for modified versions from the synchronization agent **124** (FIG. **1**), and routines for examining the version information **255** or **150** against a last synchronization signature **435** (such as a last synchronization date and time) to determine which versions have been modified. The general synchronization module **425** further includes routines for examining the version information **148** and the version information **255** or **150** to determine if one or both versions of a particular workspace element or workspace element folder have been modified.

Further, the general synchronization module **425** includes routines for performing an appropriate synchronizing responsive action. Appropriate synchronizing responsive actions may include, if only one version of a workspace element in workspace data **136** or **116** has been modified, then forwarding the modified version (as the preferred version) to the other store(s) or determining and forwarding only the changes made. Computing the changes made may be performed by examining the current status against the previous status as of the last synchronization or by comparing the two versions. It will be appreciated that no content-based review of the changes is needed. It will be appreciated that one store preferably forwards only the changes to the other store for optimizing use of processor power and minimizing the data communications across the communications channel **108** or **104**.

Other appropriate synchronizing responsive actions may include, if two versions of a workspace element have been modified independently, then instructing the content-based synchronization module **430** (described below) to execute its routines. That is, if two versions of the same workspace element have been modified independently, then a content-based review of the changes is preferable. Upon completion of the data synchronization, the general synchronization module **425** updates the last synchronization signature **435**.

The content-based synchronization module **430** includes routines for reconciling two or more modified versions of a workspace element. For example, if a user has independently modified the original and the copy of a workspace element since the last synchronization, then the content-based synchronization module **430** determines an appropriate responsive action. The content-based synchronization module **430** may request the user to select a preferred one of the modified versions or may respond based on preset preferences, i.e., by storing both versions in both stores or preferably by integrating the modified versions into a single preferred version which replaces each modified version at both stores.

The content-based synchronization module **430** examines the changes made to each version and determines if conflicts exist. When implementing version integration, a conflict may arise if inconsistent modifications such as deleting a paragraph in one version and modifying the same paragraph in the other version have been made. If a conflict exists, then the content-based synchronization module **430** attempts to reconcile the conflict, e.g., by requesting user selection or by storing both versions at both stores. Otherwise, if no conflict exists, then the content-based synchronization module **430** integrates the changes to each of the versions and updates the version information **148, 150** or **255** accordingly.

FIG. **5** is a block diagram illustrating details of the synchronization agent **124**, which includes a communications module **505** (similar to the communications module **405** described above with reference to FIG. **4**) and a general synchronization module **515** (similar to the general synchronization module **425** described above also with reference to FIG. **4**).

The communications module **505** includes routines for compressing data, and routines for communicating via the communications channel **108** with the base system **146** or via the communications channel **104** with the base system **118**. The communications module **505** may further include routines for establishing a secure communications channel through the global firewall **112** and through the LAN firewall **114** with the communications module **405**.

Similar to the general synchronization module **425**, the general synchronization module **515** includes routines for examining the version information **148** and the last synchronization signature **435** (FIG. **4**) to determine which versions have been modified and the changes made. It will be appreciated that the general synchronization module **515** may maintain its own last synchronization signature **435** copy (not shown) or may request the last synchronization signature **435** from the base system **146** or **118**. The general synchronization module **515** further includes routines for forwarding workspace data **120** determined to be modified to the general synchronization module **425**, and routines for receiving preferred versions of workspace elements of workspace data **136** or **116** or just the changes from the general synchronization module **425**.

FIG. **6** illustrates an example bookmark workspace element in the global format. The global translator **122** incorporates all the information needed by both formats (Format A and Format B) to create the Global Format. For example, if a bookmark in Format A needs elements X, Y and Z and a bookmark in Format B needs elements W, X and Y, the global translator **122** incorporates elements W, X, Y and Z to create a bookmark in the Global Format. Further, the global translator **122** incorporates the information which is needed by the synchronization means such as the last modified date. Accordingly, a bookmark in the Global Format includes a user identification (ID) **605**, an entry ID **610**, a parent ID **615**, a folder ID flag **620**, a name **625**, a description **630**, the Uniform Resource Locator (URL) **635**, the position **640**, a deleted ID flag **645**, a last modified date **650**, a created date **655** and a separation ID flag **660**.

FIG. **7** is a flowchart illustrating a method **700** for using a global translator **122** to synchronize multiple copies of a workspace element in a secure network **100**. Method **700** begins with the user interface module **410** in step **705** enabling a user to select workspace elements of workspace data **136** and **118** for the synchronization means to synchronize. The locator modules **415** in step **710** identify the memory locations of the workspace elements in workspace data **136** and **116** and the corresponding memory locations in workspace data **120**. If a selected workspace element does not have a corresponding memory location, such as in the case of adding a new workspace elements to the global server **106**, then one is selected. The selected memory location may be a preexisting workspace element or a new workspace element. As stated above, workspace element memory location identification may be implemented using intelligent software or dialogue boxes. The general synchronization module **425** and general synchronization module **515** in step **715** set the previous status of the workspace elements equal to the null set. Setting the previous status to the null set indicates that all information of the workspace element has been added.

The synchronization-start module **420** in step **720** determines whether predetermined criteria have been met which indicate that synchronization of the workspace elements selected in step **705** should start. If not, then the synchronization-start module **420** in step **725** waits and loops back to step **720**. Otherwise, the communications module **405** and communications module **505** in step **730** establish a secure communications channel therebetween.

The general synchronization module **425** and the general synchronization module **515** in step **735** determine whether any workspace elements have been modified. That is, the general synchronization module **425** in step **740** examines the version information **255** or **150** of each selected workspace element in the workspace data **136** or **116** against the last synchronization signature **435** to locate modified workspace elements. This comparison may include comparing the date of last modification with the date of last synchronization, or may include a comparison between the current status and the previous status as of the last interaction. Similarly, the general synchronization module **515** examines the version information **148** of each corresponding workspace element in workspace data **120** and the last synchronization signature **435** to locate modified workspace elements.

If in step **735** no modified workspace elements or folders are located, then the general synchronization modules **425** and **515** in step **760** update the last synchronization signature **435** and method **700** ends. Otherwise, the general synchronization module **425** in step **740** determines whether more than one version of a workspace element has been modified since the last synchronization.

If only one version has been modified, then the corresponding general synchronization module **425** or **515** in step **745** determines the changes made. As stated above, determining the changes made may be implemented by comparing the current status of the workspace element against the previous status of the workspace element as of the last interaction therebetween. If the changes were made only to the version in the workspace data **120**, then the global translator **122** in step **750** translates the changes to the format used by the other store, and the general synchronization module **515** in step **755** forwards the translated changes to the general synchronization module **425** for updating the outdated workspace element in the workspace data **136** or **116**. If the updated version is a workspace element in the workspace data **136** or **116**, then the general synchronization module **425** sends the changes to the updated version to the global translator **122** for translation and then to the general synchronization module **515** for updating the outdated workspace element in the workspace data **120**. The general synchronization module **425** and the general synchronization module **515** in step **757** update the previous state of to reflect the current state as of this interaction. Method **700** then returns to step **735**.

If the general synchronization module **425** in step **740** determines that multiple versions have been modified, then the general synchronization module **425** in step **765** computes the changes to each version and in step **770** instructs the content-based synchronization module **430** to examine content to determine if any conflicts exist. For example, the content-based synchronization module **430** may determine that a conflict exists if a user deletes a paragraph in one version and modifies the same paragraph in another version. The content-based synchronization module **430** may determine that a conflict does not exist if a user deletes different paragraphs in each version. If no conflict is found, then method **700** jumps to step **750** for translating and forwarding the changes in each version to the other store. However, if a conflict is found, then the content-based synchronization module **430** in step **775** reconciles the modified versions. As stated above, reconciliation may include requesting instructions from the user or based on preselected preferences performing responsive actions such as storing both versions at both stores. Method **700** then proceeds to step **750**.

It will be appreciated that in step **710** new workspace elements and preexisting workspace elements to which new workspace elements will be merged are set to "modified" and the previous status is set to the null set. Thus, the general synchronization module **425** in step **740** will determine that more that one version has been modified and the content-based synchronization module **430** in step **770** will determine that no conflict exists. The changes in each will be translated and forwarded to the other store. Accordingly, the two versions will be effectively merged and stored at each store.

For example, if a first bookmark folder was created by the web browser **152** on the desktop computer **134**, a second folder was created by a web browser (not shown) on the remote terminal **102**, no preexisting folder existed on the global server **106** and the user selected each of these folders for synchronization, then the synchronization means will effectively merge the first and second folders. That is, the general synchronization module **425** on the desktop computer **134** will determine that the first folder has been modified and the previous status is equal to the null set. The general synchronization module **425** will determine and send the changes, i.e., all the workspace elements in the first folder, to a new global folder on the global server **106**. Similarly, the general synchronization module **425** on the remote terminal **102** will determine that, as of its last interaction, the previous status of each of the second and the global folders is the null set. The general synchronization module **425** will instruct the content-based synchronization module **430** to examine the changes made to each folder to determine whether a conflict exists. Since no conflicts will exist, the general synchronization module **425** will forward the changes to the global folder and the general synchronization module **515** will forward its changes to the second store, thereby merging the workspace elements of the first and second folders in the global and second folders. The general synchronization module **515** will inform the general synchronization module **425** that the global folder has been modified relative to the last interaction, and will forward the new changes to the first folder. Thus, the first and second folders will be merged and stored at each store.

For a second example, the user may select an exemplary document in the LAN **110** to be synchronized. The general synchronization module **425** will forward the document to the global server **106**. Similarly, the user may select the same document for synchronization on the remote terminal **102**. The general synchronization module **515** will forward the document to the remote terminal **102**. If changes were made to the documents independently, then the content-based synchronization module **430** will examine the content of the documents to determine if a conflict exists. If no conflict exists, then as described above, the general synchronization modules **425** and **515** will merge the documents. Otherwise, if a conflict does exist, the content-based synchronization module **430** will reconcile the changes and then the general synchronization modules **425** and **515** will forward the reconciled changes to each other.

The foregoing description of the preferred embodiments of the invention is by way of example only, and other variations of the above-described embodiments and methods are provided by the present invention. For example, although the global server **106** is illustrated as a single device, the global server **106** may include several computers networked together. Components of this invention may be implemented using a programmed general purpose digital computer, using application specific integrated circuits, or using a network of interconnected conventional components and circuits. The embodiments described herein have been presented for purposes of illustration and are not intended to be exhaustive or limiting. Many variations and modifications are possible in light of the foregoing teaching. The system is limited only by the following claims.

What is claimed is:

1. A method, comprising:

providing a first memory store including a first folder for storing first record entries in a file;

providing a second memory store coupled to the first memory store, the second memory store including a second folder for storing second record entries in a file, the second folder being synchronized with the first folder at a first moment in time such that a content of the second folder is consistent with the content of the first folder at the first moment in time;

providing a global server including a third memory store having a third folder for storing third record entries in a file;

receiving new first record entries in a file at the first memory store at a second moment in time later than the first moment in time;

electronically synchronizing the first folder and the third folder, such that a new third record entry is created in the third folder based on the new first record entries; and

electronically synchronizing the third folder and the second folder, such that a new second record entry is created in the second folder based on the new third record entry.

2. A system, comprising:

a first memory store including a first folder for storing first record entries in a file;

a second memory store coupled to the first memory store the second memory store including a second folder for storing second record entries in a file, the second folder being synchronized with the first folder at a first moment in time such that a content of the second folder is consistent with the content of the first folder at the first moment in time;

an interface for receiving new record entries in a file at the first memory store;

a general synchronization module for electronically synchronizing the first folder and a third folder such that a anew third record entry is created in the third folder based on the new first record entries; and

a synchronization agent for electronically synchronizing the third folder and the second folder such that a new second record entry is created in the second folder based on the new third record entry.

3. A system, comprising:

first memory means including a first folder for storing first record entries in a file;

second memory means coupled to the first memory means, the second memory means including a second folder for storing second file entries in a file, the second folder being synchronized with the first folder at a first moment in time such that a content of the second folder is consistent with the content of the first folder at the first moment in time;

providing a global server including a third memory means having a third folder for storing third record entries in a file;

means for receiving new record file entries at the first memory means;

means for electronically synchronizing the first folder and the third folder such that a new third record entry is created in the third folder based on the new first record entries; and

means for electronically synchronizing the third folder and the second folder, such that a new second record entry is created in the second folder based on the new third record entry.

4. A method, comprising:

providing a first memory store including a first folder for storing first record entries in a file;

providing a second memory store coupled to the first memory store, the second memory store including a second folder for storing second record entries in a file the second folder being synchronized with the first folder at a first moment in time such that a content of the second folder is consistent with the content of the first folder at the first moment in time;

providing a global server including a third memory store having a third folder for storing third record entries in a file;

receiving changes at the first memory store;

electronically synchronizing the first folder and the third folder such that the content of the changes creates a third record entry; and

electronically synchronizing the third folder and the second folder such that a new second record entry is created in the second folder based on the third record entry.

5. A system, comprising:

a first memory store including a first folder for storing first record entries in a file;

a second memory store coupled to the first memory store, the second memory store including a second folder for storing second record entries in a file, the second folder being synchronized with the first folder at a first moment in time such that a content of the second folder is consistent with the content of the first folder at the first moment in time;

providing a global server including a third memory store having a third folder for storing third record entries in a file;

an interface for receiving changes at the first memory store;

a general synchronization module for electronically synchronizing the first folder and the third folder such that the content of the changes creates a third record entry; and

a synchronization agent for electronically synchronizing the third folder and the second folder such that a new second record entry is created in the second folder based on the third record entry.

6. A system, comprising:

first memory means including a first folder for storing first record entries in a file;

second memory means coupled to the first memory means the second memory means including a second folder for storing second file entries in a file, the second folder being synchronized with the first folder at a first moment in time such that a content of the second folder is consistent with the content of the first folder at the first moment in time;

providing a global server including a third memory means having a third folder for storing third record entries in a file;

means for receiving changes at the first memory means;

means for electronically synchronizing the first folder and the third folder such that the content of the changes creates a third record entry; and

means for electronically synchronizing the third folder and the second folder, such that a new second record entry is created in the second folder based on the new third record entry.

* * * * *

Exhibit 2
to
First Amended Answer to Research in Motion Limited's Complaint for Patent Infringement; Counterclaims Against Research in Motion Limited

US007228383B2

(12) **United States Patent**
Friedman et al.

(10) **Patent No.: US 7,228,383 B2**
(45) **Date of Patent: Jun. 5, 2007**

(54) **SYSTEM AND METHOD FOR PROGRESSIVE AND HIERARCHICAL CACHING**

(75) Inventors: **Gregory Scott Friedman**, Sammamish, WA (US); **Coyle Brett Marl**, Seattle, WA (US)

(73) Assignee: **Visto Corporation**, Redwood City, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/872,753**

(22) Filed: **Jun. 1, 2001**

(65) **Prior Publication Data**

US 2002/0199061 A1 Dec. 26, 2002

(51) **Int. Cl.**
G06F 13/00 (2006.01)

(52) **U.S. Cl.** 711/118; 709/217; 709/220; 709/248

(58) **Field of Classification Search** 709/206; 711/118
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,835,087 A	11/1998	Herz et al.	
6,052,735 A	4/2000	Ulrich et al.	
6,389,455 B1	5/2002	Fuisz et al.	

FOREIGN PATENT DOCUMENTS

JP	020000014	1/2000

OTHER PUBLICATIONS

M.C. Chan et al. "Application of Compaction Technique to Optimizing Wireless Email Transfer," IEEE Wireless Communications and Networking Conference 1999, p. 1535 Section A.

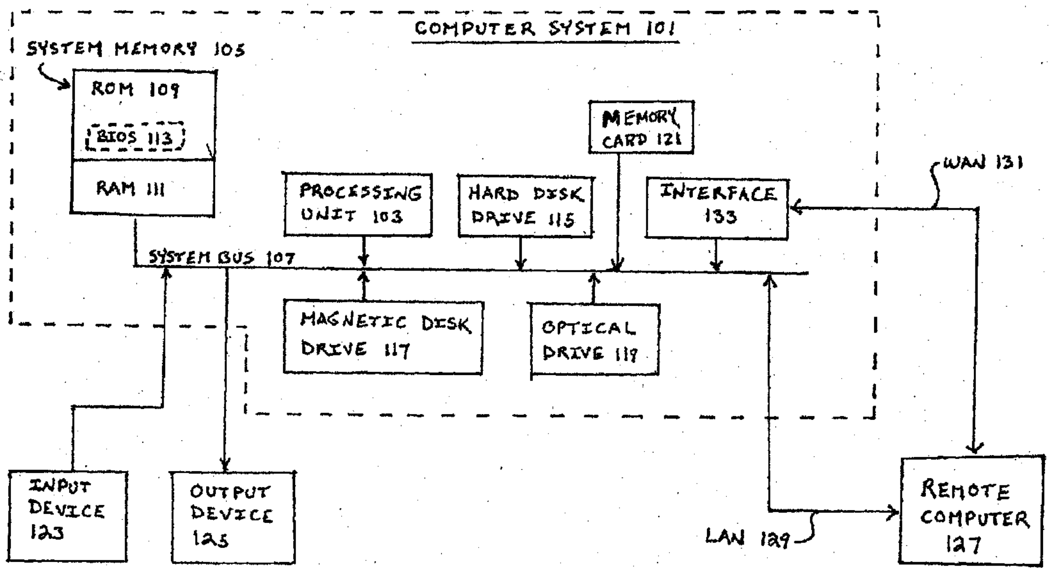
M. Crispin "Internet Message Access Protocol—Version 4 rev1," 1996 Section 2.3.1.1.

Primary Examiner—Jasmine Song
(74) *Attorney, Agent, or Firm*—Oblon, Spivak, McClelland, Maier & Neustadt, P.C.

(57) **ABSTRACT**

A system and method that allows a user to update the contents of a portable electronic mail account to correspond with the contents of a user's regular electronic mail account, where only discrete portions of the regular electronic mail account are progressively cached in the portable electronic mail account according to a hierarchy with each update operation. By updating only a portion of the regular electronic mail account with each update operation, the service provider can minimize the amount of memory storage and communication expense necessary to maintain the portable electronic mail account, while still permitting the user to access the information important to the user.

15 Claims, 4 Drawing Sheets



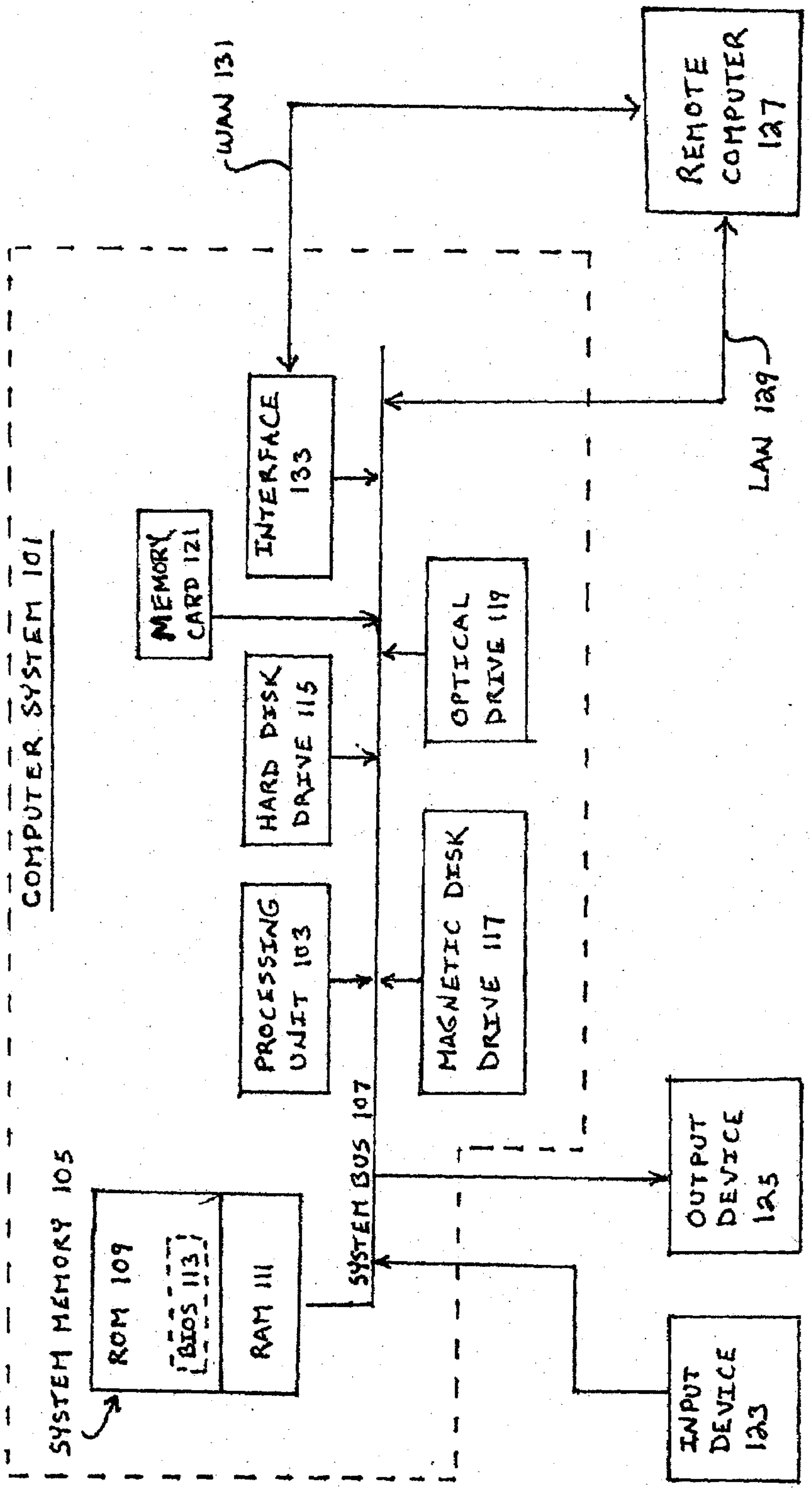

COMPUTER SYSTEM 101
SYSTEM MEMORY 105
ROM 109
BIOS 113
RAM 111
SYSTEM BUS 107
PROCESSING UNIT 103
HARD DISK DRIVE 115
MEMORY CARD 121
INTERFACE 133
MAGNETIC DISK DRIVE 117
OPTICAL DRIVE 119
WAN 131
REMOTE COMPUTER 127
LAN 129
OUTPUT DEVICE 125
INPUT DEVICE 123

FIG. 1

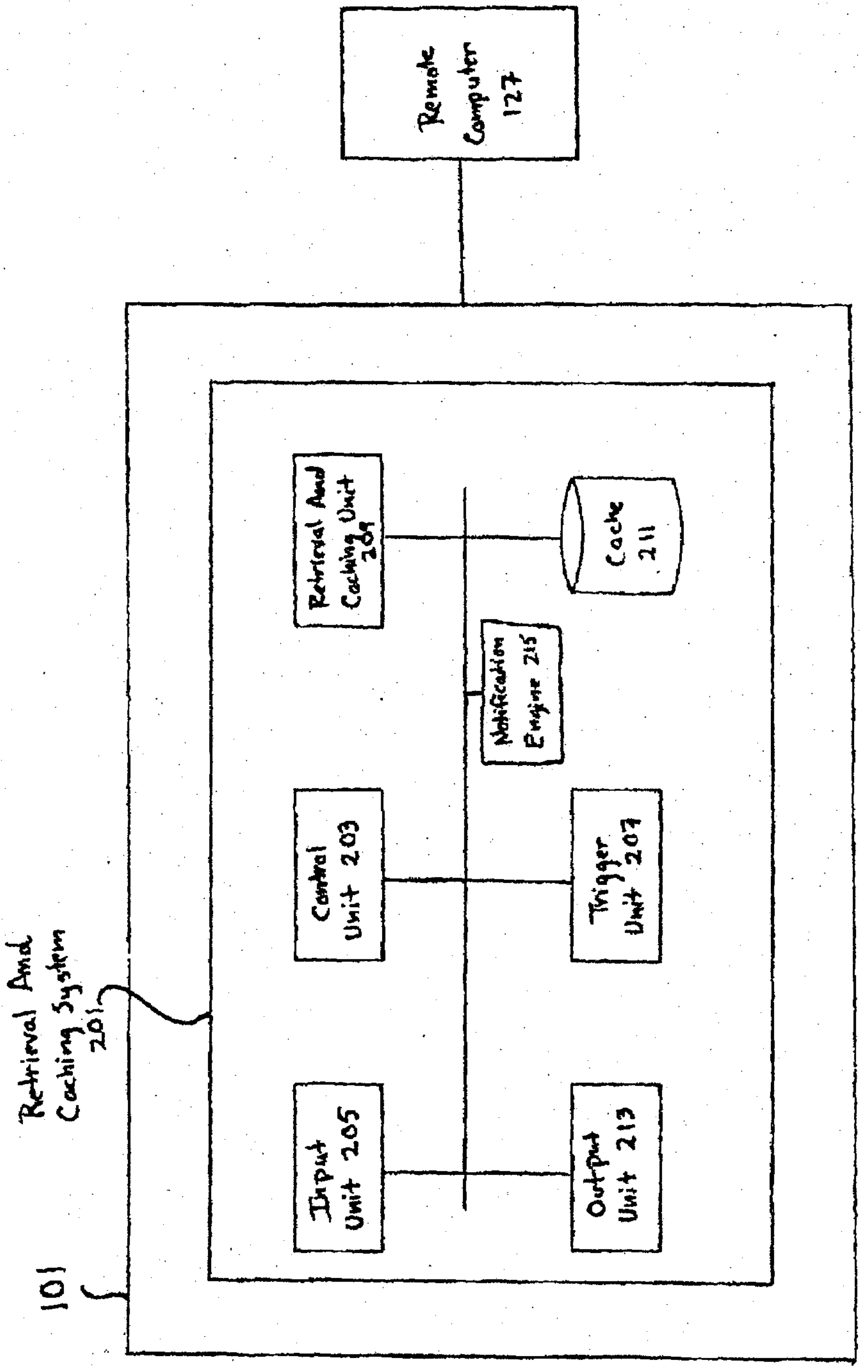
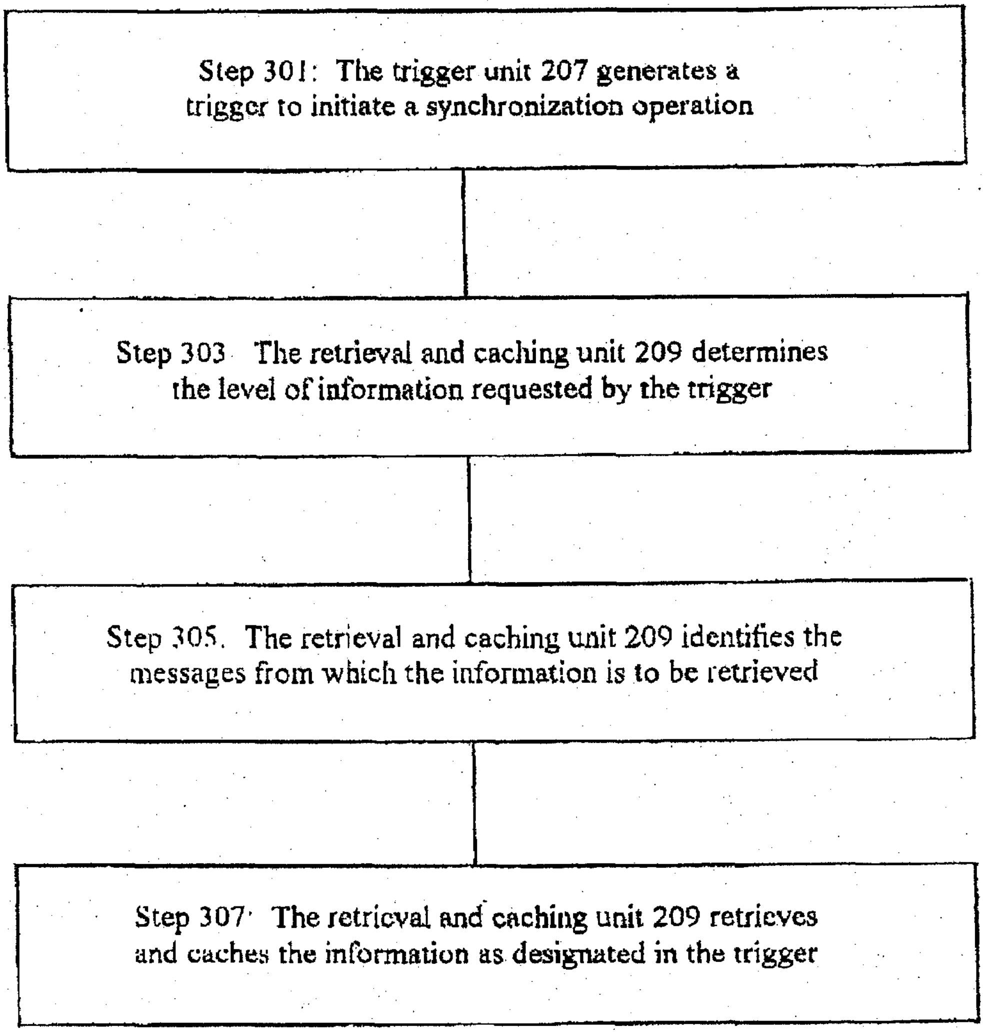

Remote Computer 127
101
Retrieval And Caching System 201
Retrieval And Caching Unit 209
Cache 211
Notification Engine 215
Control Unit 203
Trigger Unit 207
Input Unit 205
Output Unit 213

FIG 2



Step 301: The trigger unit 207 generates a trigger to initiate a synchronization operation

Step 303 The retrieval and caching unit 209 determines the level of information requested by the trigger

Step 305. The retrieval and caching unit 209 identifies the messages from which the information is to be retrieved

Step 307· The retrieval and caching unit 209 retrieves and caches the information as designated in the trigger

Fig. 3

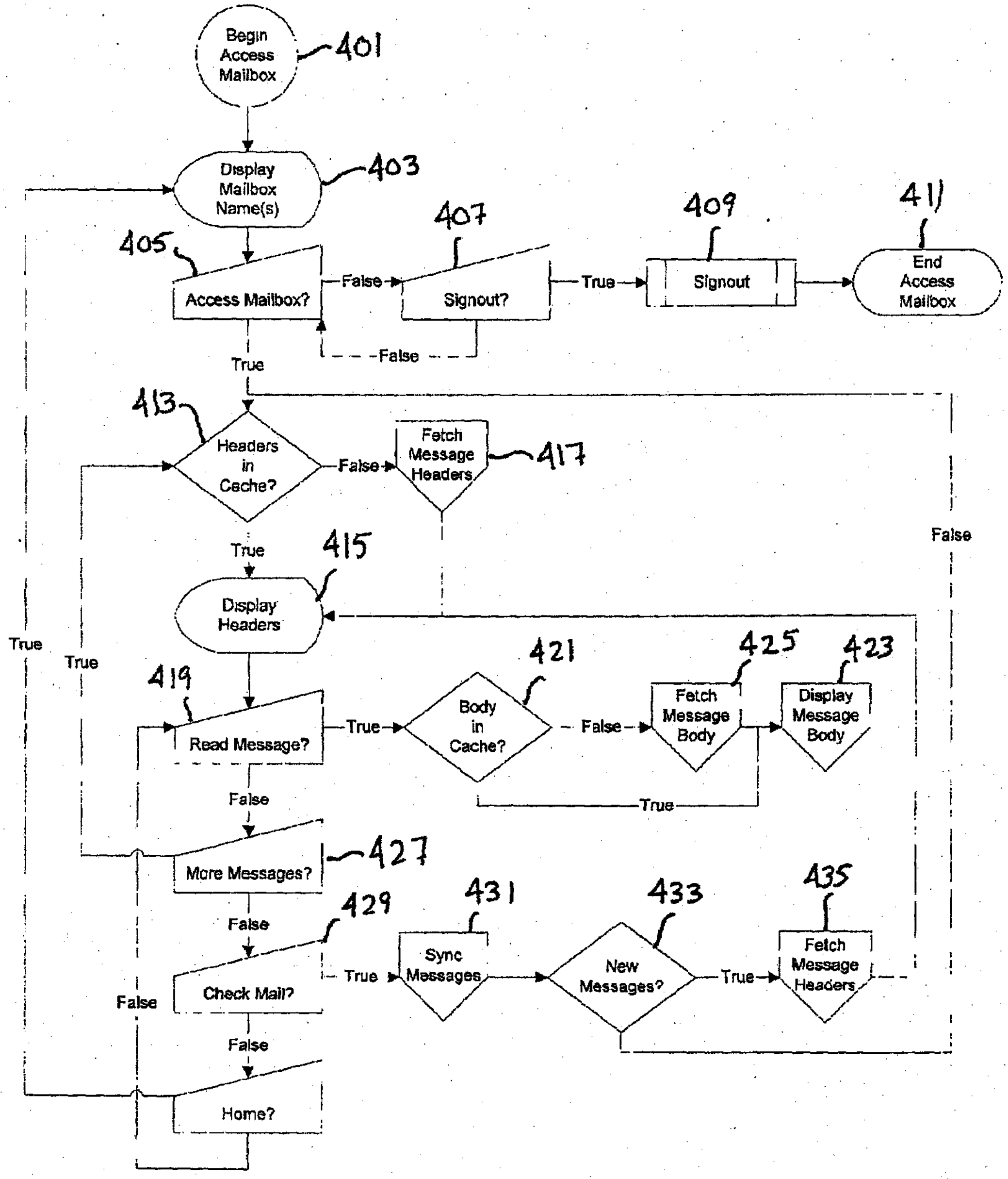

Begin Access Mailbox
401
Display Mailbox Name(s)
403
405
Access Mailbox?
False
407
Signout?
True
409
Signout
411
End Access Mailbox
False
True
413
Headers in Cache?
False
Fetch Message Headers
417
True
415
Display Headers
False
True
True
419
Read Message?
True
421
Body in Cache?
False
425
Fetch Message Body
423
Display Message Body
True
False
427
More Messages?
False
429
Check Mail?
True
431
Sync Messages
433
New Messages?
True
435
Fetch Message Headers
False
False
Home?

FIG. 4

SYSTEM AND METHOD FOR PROGRESSIVE AND HIERARCHICAL CACHING

FIELD OF THE INVENTION

The invention relates to a system and method for progressively building a cached representation, in one location, of a data set that is stored persistently in another location. More particularly, the invention relates to a method and system of periodically updating the contents of a cached representation of a data set according to a hierarchical order, so as to minimize the size of the cache, the frequency at which the cache is updated, or both.

BACKGROUND OF THE INVENTION

Electronic mail has become a commonly used and widely accepted means of communication. As a result, it is not unusual for some individuals to send and receive more than a hundred electronic mail messages in a single day. This has led to an explosion in the size of the memory required for many electronic mail accounts. For example, there are some corporate users whose mailboxes routinely exceed 50 Mbytes in size and for whom a single message with one or more attachments will often contain 5 Mbytes or more of data. Consumers also use electronic mail (or other techniques for routing electronic data files) to exchange music and other multi-media files, and these files will frequently contain multiple Megabytes of data.

As the use of electronic mail continues to grow, so does the need to access electronic mail accounts from a variety of locations. For example, many now use portable electronic communication devices for sending and retrieving electronic mail messages. These portable communication devices, which include wireless telephones, personal digital assistants, specialized portable electronic mail messaging devices, and even laptop computers, allow users to conveniently access their electronic mail accounts from almost any location, without being tied down to a stationary personal computer or network workstation.

In order to use these portable communication devices to retrieve electronic mail from one or more existing electronic mail accounts, however, the user must typically set up a new electronic mail account, specific to the device, with a service provider. The service provider must then retrieve the electronic mail files that make up the contents of the user's existing electronic mail account (i.e., the user's "regular" electronic mail account) or accounts, and store these contents in the user's electronic mail account or accounts for the portable electronic device (i.e., the user's "portable" electronic mail account or accounts). The user can then access this synchronized portable electronic mail account from the portable communication device.

Similarly, many now use aggregate electronic mail accounts to simultaneously access a variety of other existing electronic mail accounts. Again, the service provider for the aggregate electronic mail account must retrieve the contents of the user's existing regular electronic mail accounts, and synchronize these contents in the user's aggregate electronic mail account.

With a multi-tiered Web services architecture (e.g., an architecture made up of storage, networking, front-end servers and application servers), however, this synchronization process typically requires the underlying system to scale at the storage tier, which is usually the most expensive in terms of resources. Thus, in order to make this synchronization

service cost effective, the service provider will generally: (a) place a strict limit on the amount of storage memory available to the user to maintain his or her portable electronic mail account (e.g., between 2–5 Megabytes is common), or (b) charge the user a premium for the right to use more than a threshold amount of storage memory (e.g., $10–$20/month for each extra 100 Megabytes of storage memory).

When the amount of available storage is strictly capped, a user may be denied access to a particular mailbox or to a particular message within a mailbox because the storage memory cap was exceeded before all of the mailboxes or messages could be copied into the portable electronic mail account. On the other hand, where users are charged a premium for additional memory storage, the users are typically unwilling to pay an amount sufficient to cover the direct costs of such storage and/or provide a profit margin for the service provider. Accordingly, neither of these approaches is satisfactory for maintaining a portable electronic mail account.

BRIEF SUMMARY OF THE INVENTION

There is therefore a need for an electronic mail service that allows a user to employ a portable electronic mail account to more efficiently retrieve and examine the contents of one or more of his or her regular electronic mail accounts. The present invention advantageously provides a system and method that allows a user to update the contents of a portable electronic mail account to correspond with the contents of a user's regular electronic mail account, where only discrete portions of the regular electronic mail account are progressively cached in the portable electronic mail account according to a hierarchy with each update operation. By updating only a portion of the regular electronic mail account with each update operation, the service provider can minimize the amount of memory storage and communication expense necessary to maintain the portable electronic mail account, while still permitting the user to access the information important to the user.

Moreover, according to the method and system of the invention, each progressive updating step is performed in response to specific triggers. According to the invention, a trigger can be a specific request for an update operation from the user. A trigger may also be a predictive indicator, generated upon a prediction that the user desires to initiate an update operation. For example, the user may routinely initiate an update procedure to retrieve and cache the entire contents of each electronic mail message from the user's work supervisor. The method and system of the invention can then employ this routine habit of the user to predict that the user will want to initiate an update procedure to retrieve and cache the entire contents of all future electronic mail messages from the supervisor. Thus, when the electronic mail service detects an electronic mail message from the supervisor in the user's regular mailbox, the invention will generate a predictive indicator to initiate an update operation to retrieve and cache the contents of the message.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates one possible computing device for implementing various embodiments of the invention.

FIG. 2 shows a schematic embodiment of a caching device according to an embodiment of the invention.

FIG. 3 illustrates a method of updating a cache for an electronic mail accounts according to an embodiment of the invention.

FIG. 4 illustrates a method of operation for one embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

As is well known in the art, electronic mail accounts are conventionally maintained by programmable computers that are programmed to send, retrieve and store the data files that make up electronic messages. This type of computer can be embodied by, for example, an electronic mail account server. FIG. 1 shows one example of such a programmable computer system **101** capable of retrieving and caching electronic mail data files from an outside electronic mail account. The computer system **101** includes a processing unit **103**, a system memory **105**, and a system bus **107** that couples various system components, including the system memory **105**, to the processing unit **103**. The system memory **105** may include a read-only memory (ROM) **109** and a random access memory (RAM) **111**.

A basic input/output system **113** (BIOS), containing the routines that help to transfer information between elements within the computer system **101**, such as during startup, may be stored in the read-only memory (ROM) **109**. If the computer system **101** is embodied by a personal computer, it may further include a hard disk drive **115** for reading from and writing to a hard disk (not shown), a magnetic disk drive **117** for reading from or writing to a removable magnetic disk (not shown), or an optical disk drive **119** for reading from or writing to a removable optical disk (not shown) such as a CD-ROM or other optical media.

A number of program modules may be stored on the ROM **109**, the hard disk drive **115**, the magnetic disk drive **117**, and the optical disk drive **119**. A user may enter commands and information into the computer system **101** through an input device **123**, such as a keyboard, a pointing device, a touch screen, a microphone, a joystick or any other suitable interface device. Of course, the computer system **101** may employ a variety of different input devices **123**, as is known in the art. An output device **125**, such as a monitor or other type of display device, is also included to convey information from the computer system **101** to the user. As will be appreciated by those of ordinary skill in the art, a variety of output devices **125**, such as speakers and printers, may alternately or additionally be included in the computer system **101**.

In order to access electronic mail accounts, the computer system **101** preferably is capable of operating in a networked environment using logical connections to one or more remote computers, such as the remote computer **127**. The computer system **101** may be connectable to the remote computer **127** through a local area network (LAN) **129** or a wide area network (WAN) **131**, such as the Internet. When used in a networking environment, the computer system **101** may be connected to the network through an interface **133**, such as a wireless transceiver, a modem, an Ethernet connection, or any other such interface. While the interface **133** is illustrated as an internal interface in FIG. **1**, it may alternately be an external interface as is well known in the art. Of course, it will be appreciated that the network connections shown in this figure are exemplary, and other means of establishing a communications link with other computers to access an electronic mail account may be used.

An electronic mail retrieval and caching system according to one embodiment of the invention is shown in FIG. **2**. As will be discussed in detail below, the retrieval and caching system **201** progressively retrieves and caches various portions of electronic mail message data files from an electronic mail account maintained by a remote computer **127** according to a hierarchical order. The caching system **201** includes a control unit **203**, for controlling the exchange of information between the various components of the system **201**, and an input unit **205**. The input unit **205** receives information both directly from a user of an electronic mail account maintained at the remote computer **127**, and from other software applications working with the retrieval and caching system **201**. The caching system **201** further includes a trigger unit **207**, which generates triggers in response to recognizing a triggering event in information provided through the input unit **205**, as will be discussed below.

The retrieval and caching system **201** also has a retrieval and caching unit **209** and a cache **211**. In response to receiving a trigger from the trigger unit **207**, the retrieval and caching unit **209** retrieves information from one or more electronic mail accounts maintained at the remote computer **127**. It then stores the retrieved information in the cache **211**. The retrieval and caching system **201** may also include an output unit **213**, for outputting cached information to the user or to another software application working with the caching system **201**. Still further, the retrieval and caching system **201** may include a notification engine **215**. As will be discussed in detail below, the notification engine **215** periodically initiates an update operation to retrieve data into the cache **211** from the remote computer **127**.

As will be appreciated by those of ordinary skill in the art, one or more components of the retrieval and caching system **201** can be embodied using software implemented on a computer system, such as the computer system **101** shown in FIG. **2**. For example, the input unit **205**, the output unit **213**, the trigger unit **207**, the retrieval and caching unit **209** and the notification engine **215** can be embodied by software subroutines. The control unit **203** may then be a control software routine that calls the various software subroutines embodying the different units **205**–**209**, **213** and **215**. Of course, those of ordinary skill in the art will appreciate that one or more of the units **203**–**211** can also be embodied by electronic circuitry. The cache **211** may then be implemented on any suitable type of memory storage, such as the RAM **111**, the hard disk drive **115**, the magnetic disk drive **117**, or an optical disk drive **119**.

According to some preferred embodiments of the invention, the retrieval and caching system **201** operates as a read-through cache. Thus, when a user requests data from the retrieval and caching system **201**, the retrieval and caching system **201** first checks its own cache **211** for the requested data. If the cache **211** does not yet contain the requested data, then the request is "read-through" to the remote computer **127**. That is, if the cache **211** does not contain the requested data, the retrieval and caching system **201** retrieves the requested data from the remote computer **127**, and provides it to the user. The retrieval and caching system **201** also updates the contents of the cache **211** with the data retrieved from the remote computer **127** for future access by the user.

According to some embodiments of the invention, the retrieval and caching system **201** progressively updates the contents of the cache **211** with discrete categories of data from electronic mail message data files in an electronic mail account maintained on the remote computer **127** (e.g., the user's regular electronic mail account). Moreover, with some embodiments of the invention, the retrieval and caching system **201** progressively updates the contents of the cache **211** according to a hierarchical order.

For example, the retrieval and caching system 201 according to some preferred embodiments of the invention may categorize the information contained within an electronic mail message into a hierarchy with four discrete levels or tiers. The highest level of information may be an identifier for uniquely identifying each message in the remote electronic mail account. In this embodiment, the retrieval and caching system 201 will update only unique identifiers in the cache 211 with each message in the remote electronic mail account during the highest-level of the update procedure. For example, if the user's regular electronic mail account employs the commonly-used version three of the post office protocol (conventionally referred to by the acronym POP3), then the caching system 201 may update only Unique Indentification Listing (UIDL) identifier used by the POP3 protocol during an update procedure.

Thus, if the electronic mail account contains a message that is not identified in the cache 211, the retrieval and caching system 201 will add an identifier corresponding to that message to the cache 211 during the highest-level of the update procedure. Similarly, if the retrieval and caching system 201 determines during the highest level update procedure that the cache 211 contains an existing identifier for a message that has been purged from the electronic mail account, the retrieval and caching system 201 will purge that identifier (along with the electronic mail message contents corresponding to that identifier) from the cache 211. With an identifier for each message in the electronic mail account, the retrieval and caching system 201 can subsequently fetch and store other hierarchical levels of data associated with each specific electronic mail message in the remote electronic mail account, using the identifier as the key.

According to some preferred embodiments of the invention, the highest-level update procedure (i.e., the process of updating identifiers in the cache 211 to correspond with some or all of the messages in the electronic mail account) is performed for each update operation. For example, where the user's regular electronic mail account employs the POP3 protocol, each update procedure updates the identifiers in the cache 211 to ensure that the cached identifiers are current. With other preferred embodiments of the invention, however, an update procedure may perform the highest-level update of identifiers only when specifically prompted by a trigger to do so. For example, with some embodiments of the invention a user may request that the retrieval and caching system 201 retrieve only a lower hierarchical level of content (e.g., metadata, message text or attachments, as will be explained in detail below) from an electronic mail message whose identifier is already stored in the cache 211, without updating the identifiers in the cache 211.

The next highest level or tier of information for an electronic mail message may be the metadata for the message. The metadata may include, for example, the title, subject, author and size of the electronic mail message. With this categorization, for the next or second-highest level of the update procedure, the retrieval and caching system 201 may retrieve and cache only the metadata for all of the messages in the remote electronic mail account during an update operation. Alternately, the retrieval and caching system 201 may retrieve and cache the metadata for only one or more specific electronic mail messages in the second-highest level of updating during an update operation.

The information for an electronic mail message data file can be further categorized into a third tier corresponding to the entire text of the message. Thus, in the third-highest level of the update procedure, the retrieval and caching system 201 may retrieve and cache the contents of the electronic

mail messages in the user's regular electronic mail account during an update operation. Alternately, in the third-highest level of the update procedure, the retrieval and caching system 201 may retrieve and cache the contents of only specifically designated electronic mail message during an update operation.

Still a fourth category of information for an electronic mail message data file may include attachments to the message, such as word processing or spreadsheet files that are appended to the message. Thus, with this fourth-highest level of updating, the retrieval and caching system 201 may retrieve and cache the attachments appended to each of the electronic mail message data files in a user's regular electronic mail account during an update operation. Alternately, for this fourth-highest level of updating, the retrieval and caching system 201 may retrieve and cache the attachments of only specific electronic mail messages in a user's regular electronic mail account during an update operation.

As will be appreciated by those of ordinary skill in the art, the hierarchical categorization of data and corresponding update levels are merely exemplary. In fact, information contained in an electronic mail account (or any other type of electronic data file repository, such as a file location directory) can be categorized into any number of different groups as convenient. By retrieving and caching only portions of each electronic mail message according to any convenient category, the retrieval and caching system 201 need only retrieve and cache as much of each message as desired by the user. This allows the system 201 to minimize the amount of memory required to update a user's portable electronic mail account with a user's regular electronic mail account, as information that is not important to the user will not be cached.

According to still other embodiments of the invention, the retrieval and caching system 201 also progressively updates the cache 211 with an electronic mail account. That is, these embodiments of the invention minimize the number of update operations needed to update a user's portable or aggregate electronic mail account with the user's regular electronic mail account by performing an update operation only when a trigger to initiate an update operation is generated.

A method of progressively updating the cache 211 with an electronic mail account on the remote computer 127 will now be explained in detail with reference to FIG. 3. In step 301, the update operation begins when the trigger unit 207 generates a trigger to initiate the update operation. More particularly, the trigger unit 207 receives input data from the input unit 205 that represents a triggering event, by, e.g., the user that causes the trigger unit to generate a trigger. According to various embodiments of the invention, a trigger can be a requested trigger, or it can be a predictive trigger. In addition, as will be explained below, a trigger can specify or be associated with a particular level of updating for the update operation.

A variety of input data may constitute a triggering event. One such triggering event occurs when a user's portable electronic mail account requests access to the user's regular POP3 electronic mail account. In response to this request, the trigger unit 207 generates a trigger to initiate an update operation that updates the cache with the highest level of information (i.e., the identifiers for each electronic mail message data file in the user's regular electronic mail account). The retrieval and caching system 201 then updates the identifiers in the cache 211 to correspond with the messages in the user's regular electronic mail account. This trigger may also specify that another level of updating for one or more of the messages in the user's electronic mail account be included in the update operation. For example, the user's regular electronic mail account may be configured to provide the user's portable electronic mail account with the ten most recent messages for display. With this arrangement, the trigger will also specify that the update operation update the second-highest level of hierarchical information (e.g., update the metadata) for the ten most recent messages in the user's regular electronic mail account.

Yet another triggering event may occur if the user wants to view the next ten most recent messages in his or her regular electronic mail account. For example, the user may scroll down the display interface for the his or her portable electronic mail account, prompting the portable electronic mail account to request the metadata for the next ten most recent messages in the user's regular electronic mail account. If the cache **211** does not already contain these contents, then the trigger unit **207** generates a trigger specifying an update operation that updates the second-highest level of hierarchical information for the next ten most recent messages in the user's regular electronic mail account. Based upon this trigger, the retrieval and caching system **201** will update the metadata for these messages in the cache **211** with the metadata for these messages in the user's regular electronic mail account.

Still another triggering event may occur if the user requests to view the contents of a particular message displayed in the user's portable electronic mail account. If the cache **211** does not already contain these contents, then the trigger unit **207** generates a trigger specifying an update operation that updates the third-highest level of hierarchical information for the specific electronic mail message identified in the request. In response to this trigger, the retrieval and caching system **201** will then update the content for this message in the cache **211** with the content of this message in the user's regular electronic mail account.

Still yet another triggering even may occur if the user requests to view an attachment to a particular message displayed in the user's portable electronic mail account. If the cache **211** does not already contain the attachment, then the trigger unit **207** generates a trigger specifying an update operation that includes updates the fourth-highest level of hierarchical information for the specific electronic mail message identified in the request. In response to this trigger, the retrieval and caching system **201** will then update the attachment for this message in the cache **211** with the attachment of this message in the user's regular electronic mail account.

It should be noted the trigger unit **207** can be configured to generate a trigger in response to data from sources other than the user. For example, as previously noted, various embodiments of the invention may include a notification engine **215** that periodically initiates an update operation. With these embodiments, the notification engine periodically initiates the update procedure to perform a background update of the cache (i.e., to perform an update procedure without the user's intervention). For example, the notification engine **215** can advantageously be used to have the retrieval and caching system **201** periodically update the cache **211** to include identifiers and metadata for new messages in the user's electronic mail account. This allows the user to avoid the delay of performing an update operation when the user's portable electronic mail account first accesses the user's regular electronic mail account.

Still further, with various embodiments of invention, the notification engine **215** can recognize information retrieved during these periodic update procedures to trigger an update of lower-level hierarchical information from the user's regular electronic mail account. For example, a user may configure the notification engine **215** to recognize when a periodic update of the cache **211** has retrieved the identifier and metadata for an electronic mail message from a particular individual (e.g., the user's spouse), and then retrieve lower-level hierarchical information from that message. This recognition capability, sometimes referred to as filtering, is well known to those of ordinary skill in the art. When the notification engine **215** recognizes that an update of the cache **211** has retrieved identifier and metadata for a message corresponding to preset parameters (e.g., a message from a specific author, a message received during a particular time of day, a message marked with a high priority, etc.), the notification engine **215** then prompts the trigger unit **207** to retrieve a designated level of hierarchical information for that message.

Of course, those of ordinary skill in the art will appreciate that the notification engine **215** can be configured to retrieve different levels of hierarchical information for different types of messages. For example, the notification engine **215** can be set up to retrieve only message text for a message from the user's spouse, but to retrieve all levels of hierarchical information (including attachments) from messages marked as a high priority (e.g., messages marked "urgent"). Still further, various embodiments of the invention can include a notification engine **215** that provides a real time message to the user that he or she has received an electronic mail message that should be reviewed.

In addition to specific requests for an update operation from the user, the trigger unit **207** may also generate a trigger in response to predictive triggering information (i.e., the trigger unit **207** to generate a predictive trigger). For example, the user may routinely submit requests to the system **201** to retrieve and cache the entire text of all electronic mail messages authored by a particular person (e.g., the user's work supervisor). From this pattern, the trigger unit **207** may create a predictive indicator that the user will want to retrieve the entire text of all future electronic mail messages authored by that person. Accordingly, if the operation of the notification engine **215** retrieves metadata indicating that the user's regular electronic mail account has received a new message from that particular person, the trigger unit **207** may be configured to generate a predictive trigger specifying an update operation that retrieves the contents of the new message into the cache **211**.

In another example, if a user requests to view the contents of several (e.g., five) consecutive unopened messages in the user's regular electronic mail account, the trigger unit **207** may recognize that the user is simply reviewing the contents of all of that account's unopened messages in order. The trigger unit **207** may be configured to then generate a trigger that causes the system **201** to synchronize the cache **211** with the contents of the next five unopened messages on the user's regular electronic mail account.

As shown in FIG. **3**, after the trigger unit **207** issues a trigger, the retrieval and caching unit **201** initiates an update operation as specified by the trigger. First, in step **303**, the retrieval and caching unit **209** identifies the level of information requested by the trigger. As previously noted, the trigger can request a specific level of information, or simply request that the next highest level of information be retrieved and cached. In step **305**, the retrieval and caching unit **209** identifies the messages from which the information is to be retrieved. For example, the trigger may specify that information should be retrieved and cached from all of the electronic mail messages in a mailbox, just those messages sharing one or more properties (e.g., all messages received on a specific date), or a particular electronic mail message. Of course, those of ordinary skill in the art will appreciate that the order of steps **303** and **305** may be reversed.

Then, in step **307**, the retrieval and caching unit **209** retrieves the information designated in the trigger from the specified electronic mail messages that were also designated in the trigger, and caches the retrieved information in the cache **211**. From the cache **211**, the user can access and view the information using the output unit **213**.

A more detailed description of one specific embodiment of the invention will now be made with reference to FIG. **4**. As shown in this figure, in step **401** the user begins the process of accessing information contained in his or her regular electronic mail account using the retrieval and caching system **201** according to one embodiment of the invention. First, in step **403**, the system **201** displays the name of the user's regular electronic mail account. Next, in step **405**, the user may input a request to access information in that regular electronic mail account. If the user does not input a request to access the regular electronic mail account in step **405**, then the user may alternately input an instruction to sign out of the access process in step **407**. Upon receiving this instruction, a conventional sign out process is performed in step **409**, and the process of accessing the regular electronic mail account ends in step **411**.

If, however, the user submits a command to access his or her regular electronic mail account in step **405**, the retrieval and caching system **201** determines in step **413** if the cache **211** contains one or more message headers corresponding to message identifiers previously stored in the cache **211**. That is, the caching system **201** reviews at least some of the message identifiers (e.g., the highest level of hierarchical information) already stored in the cache **211**, if any, to determine if the cache **211** also contains one or more message headers (i.e., the second-highest level of hierarchal information for the electronic mail account) corresponding to these identifiers. For example, the user may typically view the first N number of electronic mail messages (e.g., ten) in the user's regular electronic mail account. The caching system **201** would then identify the first N number of unique identifiers in the cache **211**, and determine if the cache **211** also stored a message header for each of these identifiers. If the cache **211** contains these message headers, then these headers are displayed in step **415**.

On the other hand, if the cache **211** does not contain message headers corresponding sin to each of the first N number of identifiers, then the trigger unit **207** will issue a trigger instructing the retrieval and caching unit **209** to update the cache **211** with message headers corresponding to the electronic mail messages identified by at least those first N number of identifiers in the cache **211** for which the cache **211** does not already have corresponding message headers. Then, in step **417**, the retrieval and caching system **201** retrieves the electronic mail message associated with each of the first N number of unique identifiers in the cache **211**, parses the message header from each retrieved message, and stores the message header in the cache **211**. After the caching system **201** has retrieved and stored a message header for each of the first N number of identifiers in the cache **211**, the system **201** displays these headers in step **415**.

Next, in step **419**, the user may input a request to read a particular message in the user's regular electronic mail account. If the user inputs such a request, the retrieval and caching system **201** first checks in step **421** to determine if the body or text of the message selected by the user for viewing has already been stored in the cache **211**. If the body of the message has already been stored in the cache **211**, then the caching system **201** displays the message body in step **423**. If, however, the cache **211** does not contain a message body corresponding to the selected electronic mail message, then the trigger unit **207** issues a trigger instructing the retrieval and caching unit **209** to retrieve the message body for the selected message from the user's regular electronic mail account. Using the identifier for the selected message store in the cache **211**, in step **425** the retrieval and caching unit **209** then retrieves and stores the message body in the cache **211**. The retrieved message body is then displayed to the user in step **423**.

If the user does not select a message to read in step **419**, the user may alternately submit a request to view more messages from the user's regular electronic mail account in step **427**. In response to this request, the retrieval and caching system **201** returns to step **413** to display (and retrieve, if necessary) the headers corresponding to the next N number of identifiers in the cache **211**. For example, if the user typically views ten messages at a time such that N is ten, and the user has already viewed the message headers corresponding to the first ten identifiers in the cache **211** (i.e., the headers for messages **1–10**), providing a request to view more message in step **427** will cause the retrieval and caching system **201** to display (and retrieve, if necessary) the headers corresponding to the next ten identifiers stored in the cache **211** (i.e., the headers for messages **11–20**) in steps **413–417**.

If, however, the user does not input a request to review more messages in step **427**, the user may instead submit a request to check the user's regular electronic mail account in step **429**. If the user makes this request, the trigger **207** will issue a trigger to the retrieval and caching unit **209** instructing the unit **209** to update the message identifiers in the cache **211**. Thus, in step **431**, the retrieval and caching unit **209** compares the message identifiers stored in the cache **211** with the messages currently in the user's regular electronic mail account. If the electronic mail account has received new messages that do not have corresponding identifiers in the cache **211**, the retrieval and caching system **201** updates the cache **211** to include identifiers for these new messages. Similarly, if messages have been deleted from the user's regular electronic mail account, the corresponding identifier (and associated header and body information) may be deleted from the cache **211** during the update process.

Next, in step **433**, if the retrieval and caching system **201** determines that the user's regular electronic mail account has not received any new messages (i.e., each message in the user's regular electronic mail account already had a corresponding identifier in the cache **211**), then the caching system **201** returns to step **413** to display the headers for the messages (e.g., the messages identified by the current N number of message identifiers) already stored in the cache **211**. If, however, in step **433**, the caching system **201** determines that the user's regular electronic mail account has received new messages that did not already have a corresponding identifier in the cache **211**, the cache **211** is updated in step **435** to retrieve and store the header for each of these new messages.

As will be appreciated from the foregoing description, the retrieval and caching unit **209** allows the information in a user's portable electronic mail account, maintained in cache **211**, to be progressively updated with information from the user's regular electronic mail account. Moreover, the update procedure can be performed only to a degree that is desired by the user. This allows the retrieval and caching unit **209** to minimize the amount of the cache **211** required to update the two electronic mail accounts. It also allows the retrieval and caching unit **209** to minimize the number of update operations, as these operations may only be performed in response to a trigger. Further, by minimizing the required storage memory, the information can be cached in a fast semiconductor RAM memory, rather than in a slower, large-scale read/write memory such as a magnetic or optical disk drive. This significantly improves the speed at which the user can access the information stored in the cache.

By thus progressively obtaining and caching only a portion of the relevant information for the electronic mail messages in each retrieval and caching operation, the caching method and system according to the invention employs only the minimum amount of storage memory and network communication expense required to obtain the information needed by the user.

It should be noted that, while the above-described embodiments of the invention are described primarily with regard to the efficient retrieval and caching of message information from an electronic mail account, those of ordinary skill in the art will appreciate that the invention can be employed with any data set for which a representation will be maintained at a different location. Thus, the invention may be employed to cache information relating to any type of electronic file, and is not limited to electronic mail message files.

For example, the invention may be employed to progressively and hierarchically update a file directory, such as that provided by Windows Explorer, for data files maintained at a different storage location. This use of the invention may be beneficial, for example, where a user employs a client computer to access and manipulate data files maintained at a remote server computer. Thus, the invention may be employed to progressively retrieve different hierarchical levels of information associated with the data files (e.g., an identifier, metadata, file content, text content, image content, etc.) into a local memory on the user's client machine in the manner described above.

With this arrangement, a user may only selectively review, e.g., portions of a directory containing word processing files. Moreover, the retrieval and caching system **201** of the invention could be configured to send operational instructions to the remote computer **127** where the electronic files are persistently stored. Thus, instead of reviewing the entire directory listing, the user could view and traverse a file directory structure containing only a portion of the information for the file directory, and perform useful transactions on individual files, such as printing, faxing, forwarding, sharing, and deleting, without having to retrieve and store additional information, such as the contents of the electronic file itself, at the user's client computer.

The present invention has been described above by way of specific exemplary embodiments, and the many features and advantages of the present invention are apparent from the written description. Thus, it is intended that the appended claims cover all such features and advantages of the invention. Further, since numerous modifications and changes will readily occur to those skilled in the art, the specification is not intended to limit the invention to the exact construction and operation as illustrated and described. For example, the invention may include any one or more elements from the apparatus and methods described herein in any combination or subcombination. Accordingly, there are any number of alternative combinations for defining the invention, which incorporate one or more elements from the specification (including the drawings, claims, and summary of the invention) in any combinations or subcombinations. Hence,

all suitable modifications and equivalents may be considered as falling within the scope of the appended claims.

What is claimed is:

1. A method of caching information relating to a set of data items, comprising:

providing a first memory storing a set of data items; wherein information of each of the data items is categorized into hierarchical levels;

in a first retrieval operation, retrieving from the first memory and storing into a second memory only a first hierarchical level of information corresponding to at least one of the data items, wherein the first hierarchical level of information comprises at least one identifier for uniquely identifying each of the at least one of the data items; and

in a second retrieval operation separate from the first retrieval operation, retrieving from the first memory and storing into the second memory only a second hierarchical level of information corresponding to the at least one of the data items, wherein the second hierarchical level of information comprises metadata for the at least one of the data items.

2. The method of caching information recited in claim **1**, wherein the data items are electronic data files.

3. The method of caching information recited in claim **2**, wherein the electronic data files are electronic mail message data files.

4. The method of caching information recited in claim **1**, wherein the at least one identifier for uniquely identifying each of the at least one of the data items is selected to comprise Unique Identification Listing (UIDL) identifier.

5. The method of caching information recited in claim **1**, wherein the metadata includes one or more data selected from the group consisting of: a title of the at least one of the data items, a subject of the at least one of the data items, an author of the at least one of the data items, and a size of the at least one of the data items.

6. The method of caching information recited in claim **1**, wherein the first memory corresponds to a user's regular electronic mail account, and the second memory corresponds to a user's portable electronic mail account.

7. The method of caching information recited in claim **1**, wherein the data items are electronic mail messages, and further comprising:

in a third retrieval operation separate from the first and second retrieval operations, retrieving and storing into the memory only a third hierarchical level of information corresponding to the at least one of the electronic mail messages.

8. The method of caching information recited in claim **7**, wherein the third level of hierarchical information is an attachment to the at least one of the electronic mail messages.

9. The method of caching information recited in claim **7**, wherein the third level of hierarchical information is text content of the body of the at least one of the electronic mail messages.

10. A method of caching information relating to a set of data items, comprising:

in a first retrieval operation, retrieving and storing into a memory only a first hierarchical level of information corresponding to at least one of the data items; and

in a second retrieval operation separate from the first retrieval operation, retrieving and storing into the memory only a second hierarchical level of information corresponding to the at least one of the data items, wherein the first retrieval operation is initiated in response to receiving a first trigger, and the second retrieval operation is initiated in response to receiving a second trigger different from the first trigger; and

wherein at least one of the first trigger and the second trigger is a predictive indicator predicting a user's preferences for retrieving information from the set of data items.

11. The method of caching information recited in claim **10**, wherein the set of data items are stored in a memory associated with a user's regular electronic mail account, and the first and second retrieval operations comprise storing the first and second hierarchical levels of information in a memory associated with a user's portable electronic mail account.

12. The method of caching information recited in claim **10**, wherein the user's preference for retrieving information is based on retrieving electronic mail messages authored by a particular person.

13. The method of caching information recited in claim **10**, wherein the user's preference for retrieving information is based on a number of unopened electronic mail messages.

14. The method of caching information recited in claim **10**, wherein the data items are file directory information.

15. The method of caching information recited in claim **10**, wherein the data items are electronic mail message data files.

* * * * *